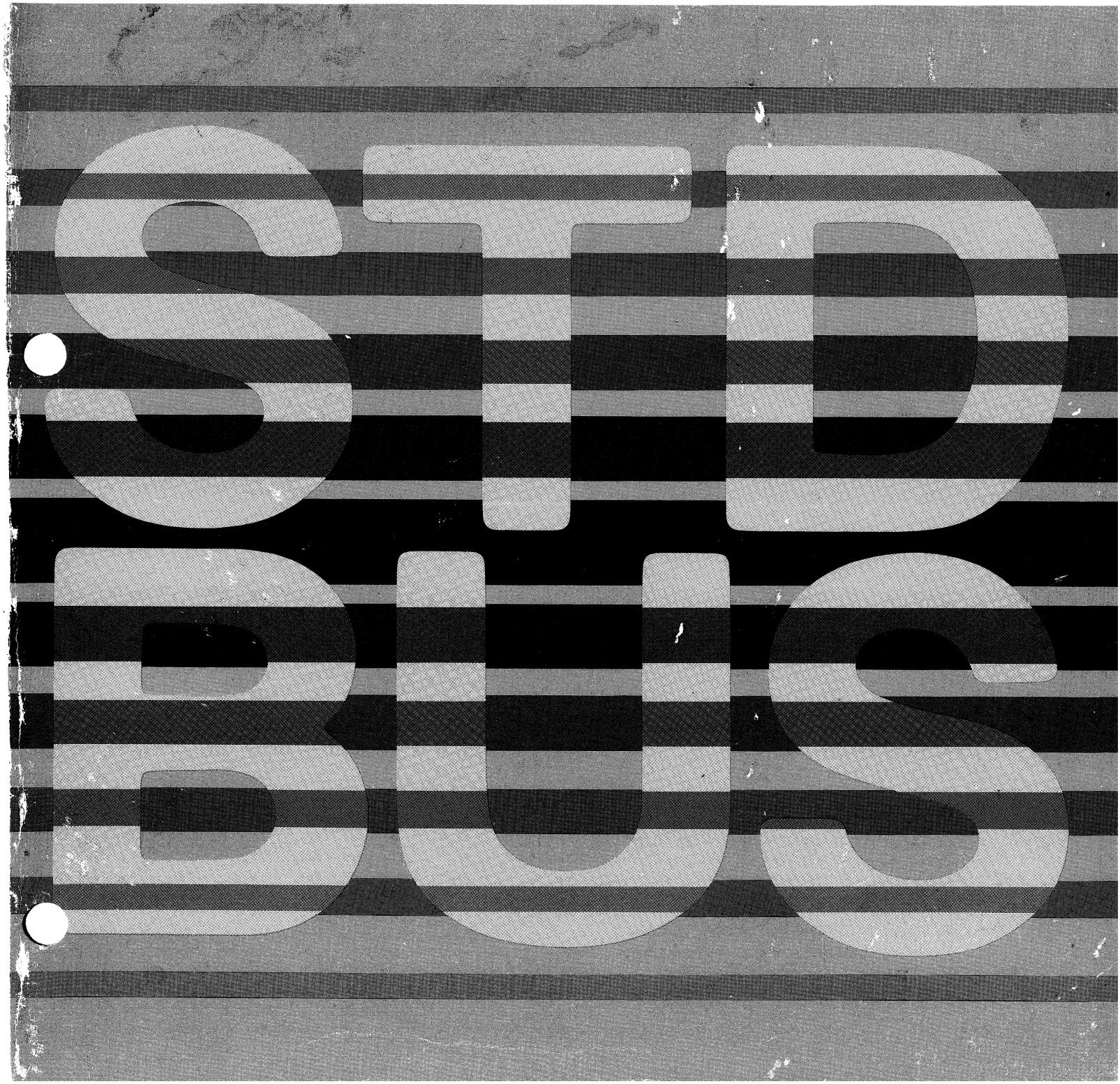
PRO-LOG®
CORPORATION

STD 7000

7806/7807
Z80A Multifunction
CPU Cards

USER'S MANUAL



NOTICE

The information in this document is provided for reference only. Pro-Log does not assume any liability arising out of the application or use of the information or products described herein.

This document may contain reference information and products protected by copyrights or patents and does not convey any license under the patent rights of Pro-Log, nor the rights of others.

Printed in U.S.A. Copyright 1984 by Pro-Log Corporation, Monterey, CA 93940. All rights reserved. However, any part of this document **may be reproduced** with Pro-Log Corporation cited as the source.

STD 7000

7806/7807

Z80A Multifunction CPU Cards

USER'S MANUAL

A Pro-Log OPENWARE ™ Product

FOREWORD

This manual explains how to use the Pro-Log 7806 and 7807 Z80A Multifunction CPU Cards. It is structured to reflect the answers to basic questions you might ask yourself about the 7806 or 7807. We welcome your suggestions on how we can improve our instructions.

The 7806 and 7807 cards are part of a family of cards and have a great deal in common. These cards are combined in this manual because of this commonality and to make it easier for the user to take advantage of this family and tailor the card to his/her application. The term 7806/7 will be used in all statements that apply to both cards.

The 7806/7807 family is part of the Pro-Log Series 7000 STD BUS hardware. Our products are modular, and designed and built with second-sourced parts that are industry standards. They provide industrial managers with the means of using their own people to control the design, production, and maintenance of the products that use STD BUS hardware.

Pro-Log supports its products with thorough and complete documentation. Also, we teach courses on how to design with and use microprocessors and the STD BUS products.

You may find the following Pro-Log documents useful in your work: the Microprocessor User's Guide and the STD BUS Specification and Practice. If you would like a copy of these documents, please write to us on your company letterhead.

CONTENTS

	Page
Foreword	ii
Figures	v
Quick Reference Aids	viii
References/Bibliography	viii
Section 1. Purpose and Main Features	1-1
Purpose	1-1
Features	1-1
Section 2. Installation and Specifications	2-1
Introduction	2-1
Installation	2-1
Card Layout and Jumpers	2-4
Cabling	2-4
Serial Channel	2-5
Memory	2-6
Selecting Memory Mapping	2-6
Configuring the Memory Sockets	2-6
Disabling Unused Sockets	2-6
MEMEX	2-9
Segment	2-9
Memory Decoder PROM	2-10
I/O Mapping	2-10
I/O Decoder	2-12
IOEXP	2-13
Priority Chain	2-13
Specifications	2-14
Mechanical Specifications	2-14
Reset	2-14
Section 3. Operation	3-1
Introduction	3-1
Memory Expansion	3-1
Serial Channel and Parallel Handshake	3-2
DART Internal Structure	3-2
Explanation of Registers	3-3
DART Operation	3-11
7807 Parallel Port	3-16
Counter/Timer	3-16
Control Registers	3-18
Load Register	3-19
Current Count Register	3-19
Vector Register	3-19
CTC Operation	3-19

Page

Section 4. Operating Software **4-1**

Introduction 4-1
Memory Addresses 4-1
I/O Port Adresses 4-1
Description 4-1
7806 Subroutine Specifications
(INIT COMM) 4-4
(INIT INTER) 4-5
(INIT CTC 0) 4-5
(INIT CTC 1) 4-6
(INIT DART) 4-6
(INIT DART CH) 4-7
(STRT XMT A) 4-7
(STRT XMT B) 4-8
(RCV & ECHO A) 4-9
(RCV & ECHO B) 4-10
7807 Subroutine Specifications
(INIT SER COMM) 4-29
(INIT INTER) 4-30
(INIT CTC 0) 4-30
(INIT PAR PORT) 4-31
(INIT DART CH) 4-32
(STROBE*) 4-32
(ENABLE HNDSHK) 4-33
(BUSY?) 4-33
(PAPER END?) 4-34
(STRT XMT B) 4-34
(PRINT DATA A) 4-35
(RCV & ECHO B) 4-36

Section 5. Maintenance **5-1**

Reference Drawings 5-1
Return for Repair/Warranty 5-12

Appendix A. Quick Reference **A-1**

Introduction A-1

FIGURES

Figure		Page
1-1	7806 Z80A Multifunction CPU Card	1-2
1-2	7806 Block Diagram	1-2
1-3	7807 Z80A Multifunction CPU Card	1-3
1-4	7807 Block Diagram	1-3
2-1	Installation	2-1
2-2	Physical Layout and Jumper Options	2-2
2-3	Cables for Use with 7806/7	2-4
2-4	Serial Port DCE/DTE Jumpers	2-5
2-5	Memory Mapping Options	2-7
2-6	Memory Socket Option Jumpers	2-8
2-7	Decoder PROM Part Numbers and Programming Equipment	2-10
2-8	Memory Decoder PROM Programming Information	2-11
2-9	Memory Decoder PROM Example, Option #0	2-12
2-10	I/O Decoder PROM Data	2-13
2-11	Opcode Fetch and Refresh Timing	2-15
2-12	Memory Read Timing Other Than Opcode Fetch	2-16
2-13	Memory Write Timing	2-16
2-14	Output Timing	2-17
2-15	Input Timing	2-17
2-16	Bus Request Timing	2-18
2-17	Interrupt Request Timing	2-18
2-18	Nonmaskable Interrupt Request Timing	2-19
2-19	Wait State Timing	2-19
2-20	Electrical and Environmental Specifications	2-20
2-21	Serial Switching Levels	2-20
2-22	7807 Parallel Connector J1 Pin List	2-21
2-23	Serial Connector Pin List	2-22
2-24	Counter/Timer Connector J3 Pin List	2-22
2-25	Segment Connector J4 Pin List	2-23
2-26	STD BUS Connector Pin List	2-23
3-1	I/O Port Functions	3-1
3-2	DART Internal Structure	3-2
3-3	DART Register Functions, General	3-3
3-4	DART Register Functions, Detailed	3-4
3-5	Baud Rate Selection	3-12
3-6	Interrupt Vectors and Priority	3-13
3-7	DART Handshake Capability	3-15
3-8	CTC Channels Per Function	3-16
3-9	CTC Internal Structure	3-17
3-10	CTC Control Registers	3-18
7806 Software		
4-1	7806 Subroutine Types	4-2
4-2	Main Subroutines Use Requirements	4-3
4-3	Initialization Values	4-4
4-4	Program Memory Map	4-11
4-5	RAM Memory Map	4-12

Figure **Page**

4-6 (INIT COMM) Flowchart 4-13
4-7 (INIT CTC 0) Flowchart 4-13
4-8 (INIT CTC 1) Flowchart 4-13
4-9 (INIT INTER) Flowchart 4-13
4-10 (INIT DART) Flowchart 4-14
4-11 (INIT DART CH) Flowchart 4-14
4-12 (STRT XMT A) Flowchart 4-15
4-13 (STRT XMT B) Flowchart 4-15
4-14 (XMT BLK A) Flowchart 4-15
4-15 (XMT BLK B) Flowchart 4-15
4-16 (XMT CHAR) Flowchart 4-16
4-17 (END XMT?) Flowchart 4-16
4-18 (XMT OFF) Flowchart 4-16
4-19 (RCV & ECHO A) Flowchart 4-17
4-20 (RCV & ECHO B) Flowchart 4-17
4-21 (RCV & ECHO) Flowchart 4-17
4-22 Software Listing 4-18

7807 Software

4-23 7807 Subroutine Types 4-26
4-24 Main Subroutines Use Requires 4-28
4-25 Serial Initialization Values 4-29
4-26 Parallel Initialization Values 4-31
4-27 Program Memory Map 4-37
4-28 RAM Memory Map 4-38
4-29 (INIT SER COMM) Flowchart 4-39
4-30 (INIT CTC 0) Flowchart 4-39
4-31 (INIT PAR PORT) Flowchart 4-39
4-32 (INIT INTER) Flowchart 4-39
4-33 (STROBE*) Flowchart 4-40
4-34 (INIT DART CH) Flowchart 4-40
4-35 (ENABLE HNDSHK) Flowchart 4-41
4-36 (STRT XMT B) Flowchart 4-41
4-37 (BUSY?) Flowchart 4-41
4-38 (PAPER END?) Flowchart 4-41
4-39 (XMT BLK B) Flowchart 4-42
4-40 (XMT CHAR) Flowchart 4-42
4-41 (END XMT?) Flowchart 4-43
4-42 (XMT OFF) Flowchart 4-43
4-43 (PRINT DATA A) Flowchart 4-43
4-44 (RCV & ECHO B) Flowchart 4-44
4-45 (RCV & ECHO) Flowchart 4-44
4-46 (PRINT DATA) Flowchart 4-45
4-47 Software Listing 4-46

5-1 7806 Schematic 5-2
5-2 7806 Assembly Drawing 5-6
5-3 7807 Schematic 5-7
5-4 7807 Assembly Drawing 5-11

Figure **Page**

A-1 DART Register Functions, General A-1
A-2 DART Register Functions, Detailed A-2
A-3 Serial Port DCE/DTE Jumpers A-3
A-4 Baud Rate Selection A-4
A-5 I/O Port Functions A-5
A-6 Interrupt Vectors and Priority A-5
A-7 CTC Control Registers A-6
A-8 7806 Main Subroutines Use Requirements A-6
A-9 7807 Main Subroutines Use Requirements A-7

QUICK REFERENCE AIDS

This manual is designed to instruct the new user of the 7806/7 card and to be a quick source of information for those familiar with it. New users are encouraged to read all sections pertaining to their application before operating the 7806/7. This will help familiarize the user with all of the card's capabilities and all aspects of the card's operation which affect the application. Once familiar with the 7806/7, pertinent data can be found easily by using the "Quick Reference" section at the back of this manual.

REFERENCES/BIBLIOGRAPHY

Mostek 1979 Microcomputer Data Book. Mostek Corporation, 1979.

Mostek 1982/1983 Microelectronic Data Book. Mostek Corporation, 1982.

1982/83 Data Book. Zilog, Inc., 1982.

Microprocessor Applications Reference Book, Volume 1. Zilog, Inc., 1981

NOTE: The Z80-DART is functionally identical to the Z80-SIO in asynchronous applications; some data sheets used were for the Z80-SIO.

PURPOSE

The 7806/7 Z80A Multifunction CPU Cards combine a 4 MHz Z80A processor with a 4-channel counter/timer and four 28-pin memory sockets on one STD BUS card. The 7806 also features two serial RS-232-C channels, while the 7807 has one serial channel and an 8-bit parallel output port with handshake.

The RS-232-C port(s) can function as data terminal or data communications equipment and has full handshake capability. The parallel port can function as an interface to Centronics-type printers or as a general-purpose output port with handshake. The counter/timer has four timer channels, three of which can be used as external event counters or as prioritized interrupt inputs with vectors. One of the channels can be used to generate baud rates for the RS-232-C port. The four memory sockets can accept byte-wide RAMs (2 or 8K bytes) or EPROMs (2 to 32K bytes). This combination of functions on one card greatly reduces the card count and cost of an STD BUS system.

FEATURES

- Fully buffered and expandable Z80A processor
- Three versions: 4 MHz (7806/7-0), 3.6864 MHz (7806/7-1), and 2.5 MHz (7806/7-2)
- Control of STD BUS memory expansion line
- Capacity for up to 128K of EPROM or 32K of RAM onboard
- Highly flexible memory mapping
- RS-232-C channel(s) with full handshake capability (7806 has 2 channels, 7807 has 1 channel)
- Modem interface compatibility
- Baud rates to 38400 with 3.6864 MHz (7806/7-1) or 2.5 MHz (7806/7-2)
- Eight-bit parallel output port with handshake (7807 only)
- Four 8-bit timer channels
- Three timer channels for use as external event counters or prioritized, vectored, interrupt inputs
- Bootstrap EPROM capability
- DMA to onboard memory capability
- Responds to Segment to participate in large memory schemes

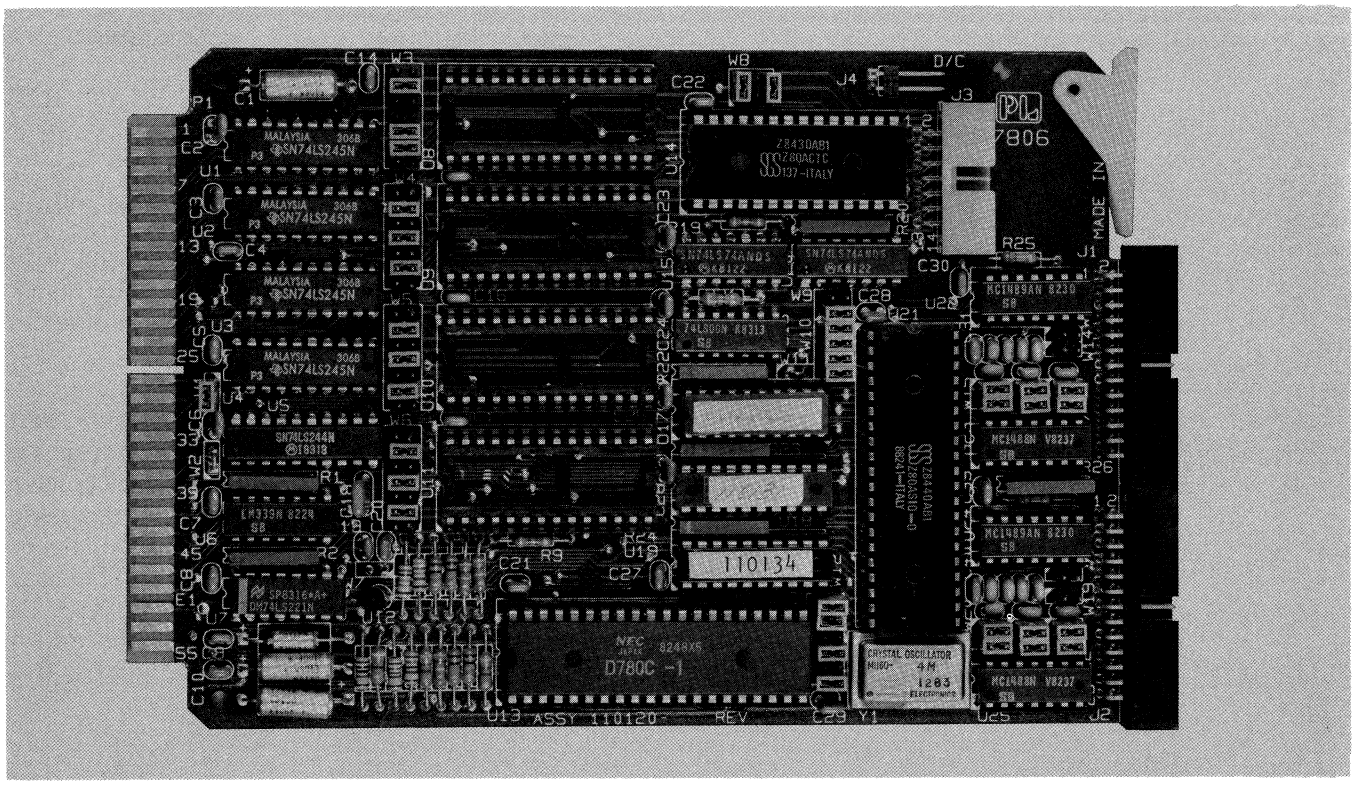


Figure 1-1. 7806 Z80A Multifunction CPU Card

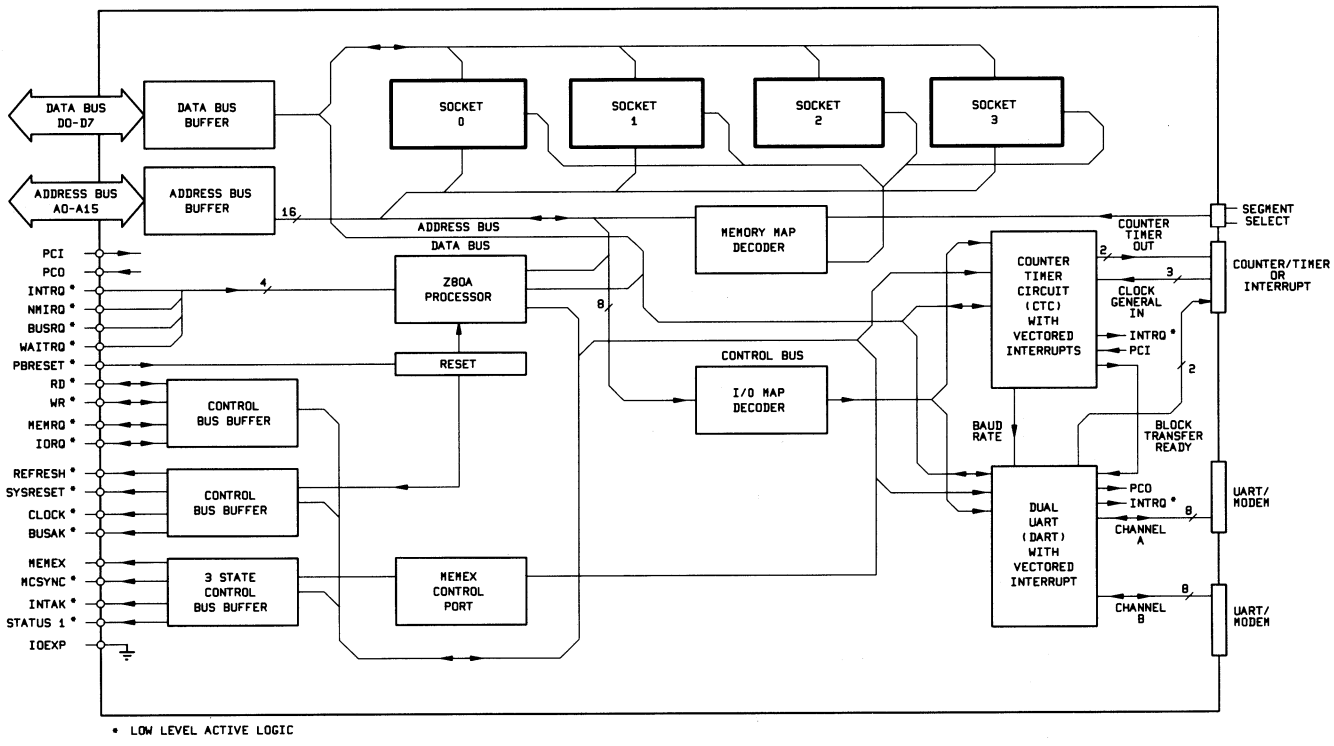


Figure 1-2. 7806 Block Diagram

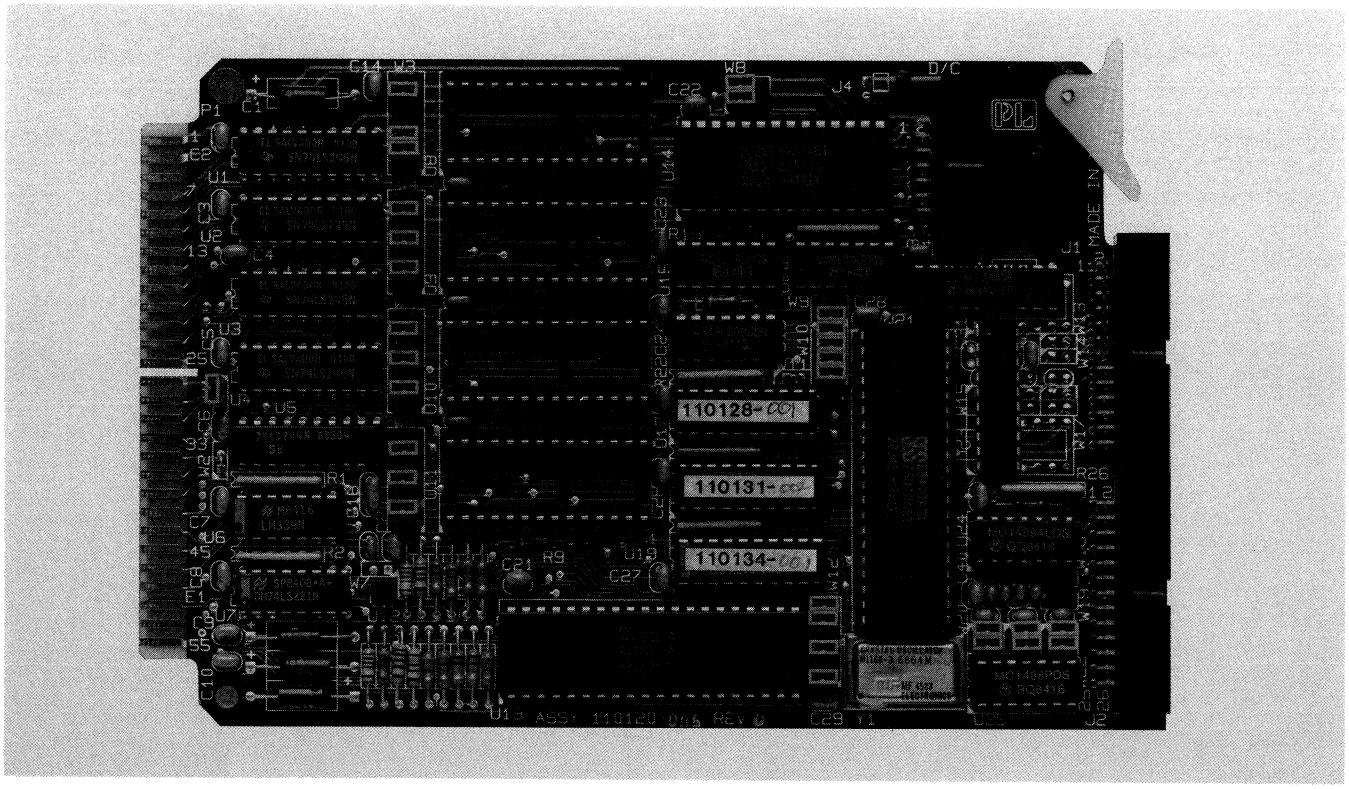


Figure 1-3. 7807 Z80A Multifunction CPU Card

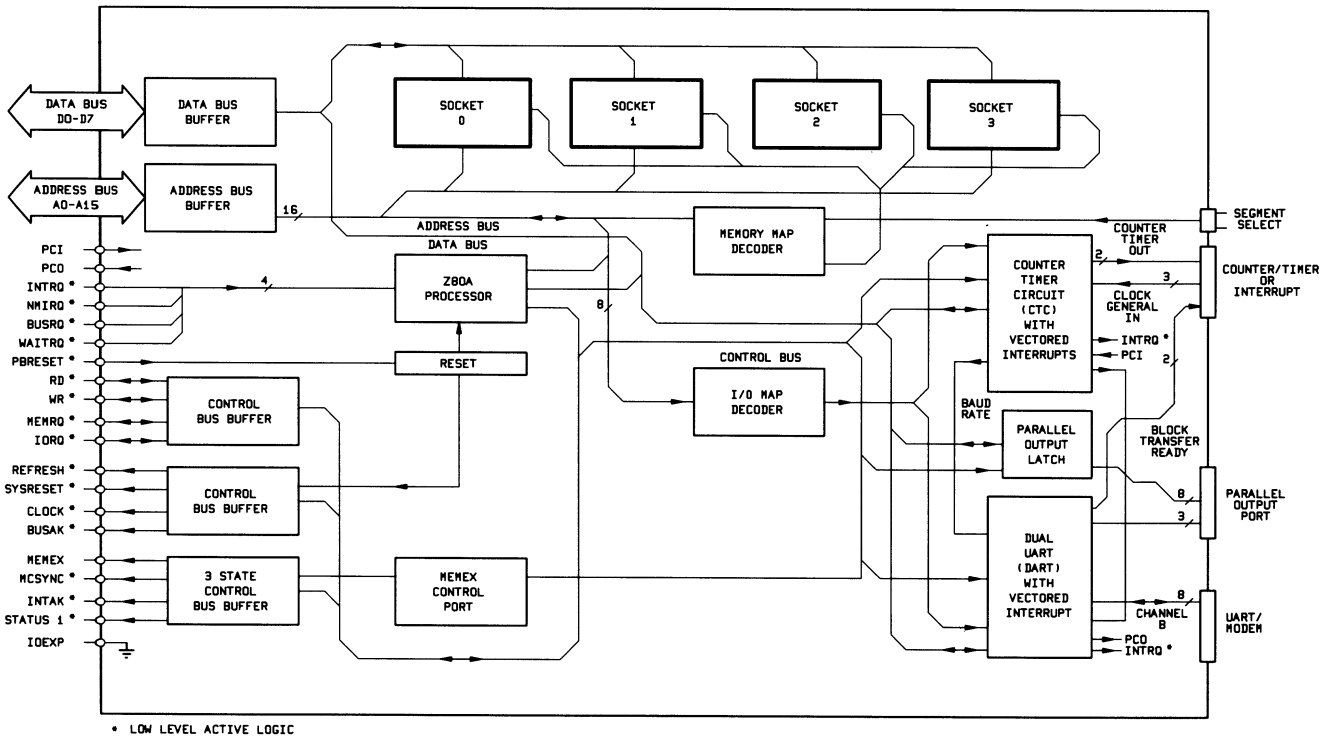


Figure 1-4. 7807 Block Diagram

INTRODUCTION

This section covers all aspects of the 7806/7 card which may need to be reviewed before operation, including optional functions selected by shorting plugs and cables to be used.

INSTALLATION

> **CAUTION**
>
> To prevent possible damage to your STD BUS system, make sure power is **off** before inserting a card into the card rack or before removing a card from the card rack.

The 7806/7 can operate independently or as part of an STD BUS system. If used as part of an STD BUS system, it should be installed in an STD BUS backplane, or card rack, with the card ejector toward the top as shown in figure 2-1. It can be installed in any slot in the rack. If access to the card is needed for testing or servicing, a card extender such as the Pro-Log model 7901 may be used.

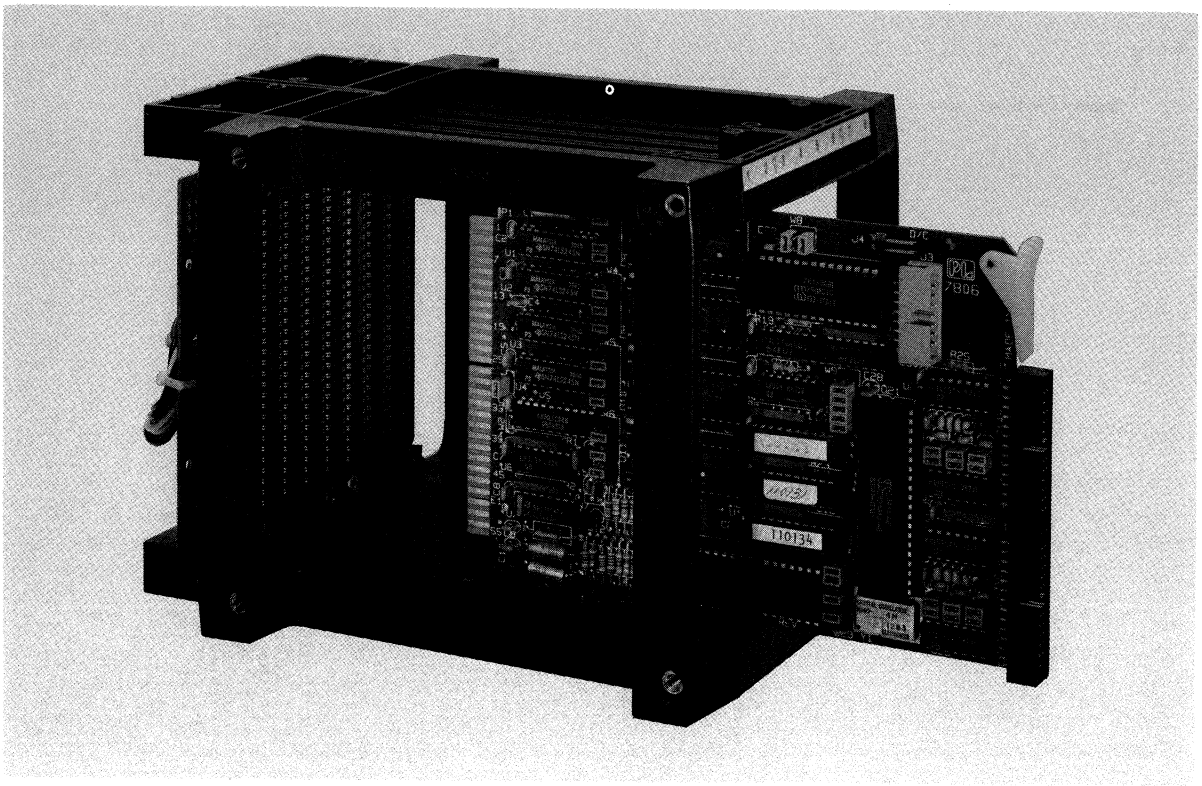


Figure 2-1. Installation (7806/7)

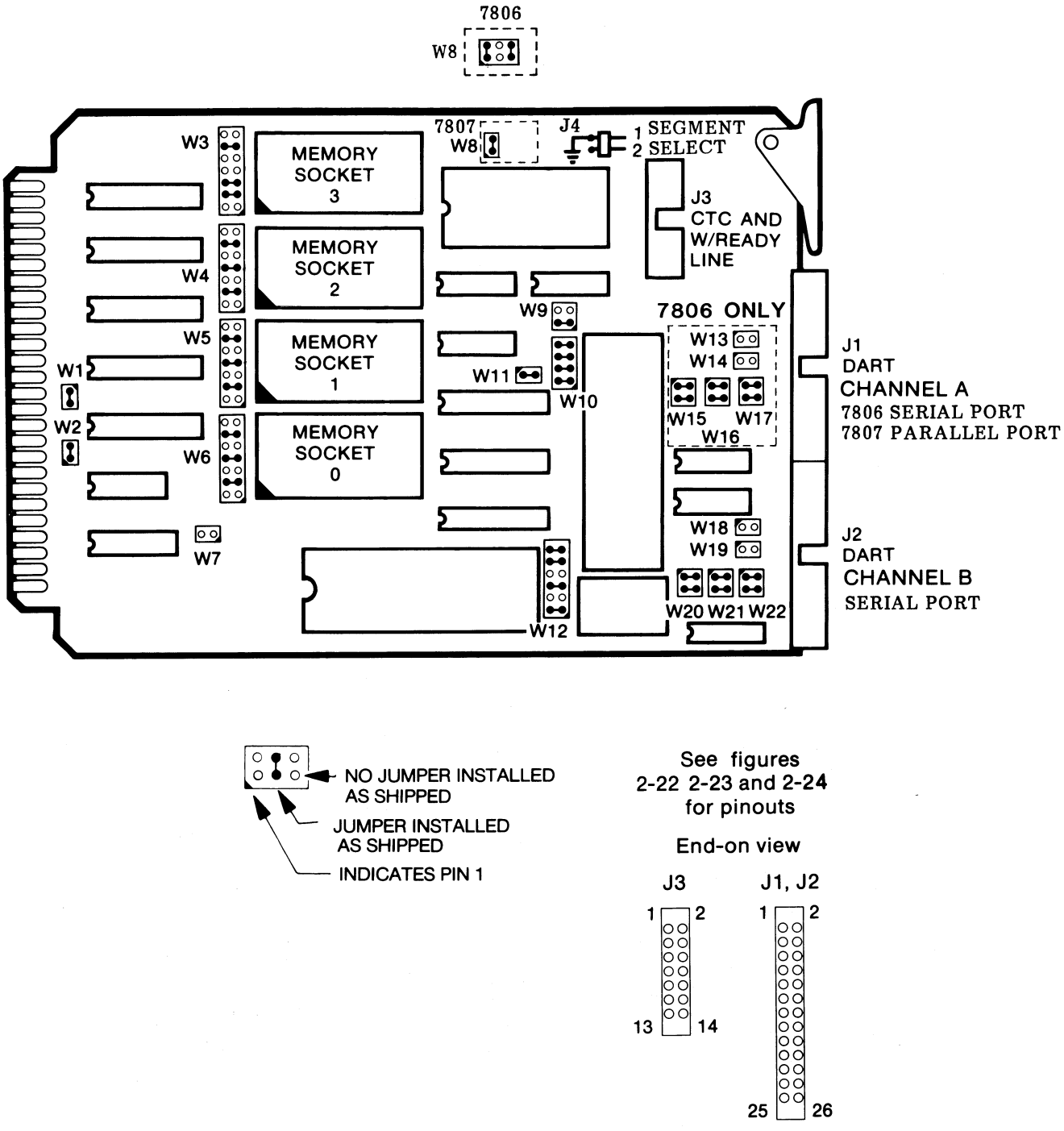


Figure 2-2. Physical Layout and Jumper Options (7806/7)

Jumper Group	DESCRIPTION	As Shipped
W1	1-2 7806/7 controls **MEMEX** none MEMEX floating, must be controlled by another card.	1-2
W2	1-2 **IOEXP** connected to ground none IOEXP floating, must be controlled by another card.	1-2
W3-W6	Configure **Memory Sockets** 3-0 respectively for memory size and ROM or RAM. See figure 2-6.	W3 3-4, 5-6, 11-12 W4-W6 3-4, 7-8, 11-12
W7	For Pro-Log use only.	
W8	**7806:** 1-2 CTC channel 1 generates baud clock for 5-6 DART channel B. 3-4 CTC channel 0 generates baud clock for both DART channels. **7807:** 1-2 Onboard clock ÷2 input to CTC channel 1. none External input or system clock input to CTC channel 1.	**7806:** 1-2 5-6 **7807:** 1-2
W9	1-2 **MEMEX** low at power-up and MEMEX port noninverting. 3-4 MEMEX high at power-up and MEMEX port inverting.	1-2
W10	1-2 thru 7-8 Enable Memory Sockets 3-0 respectively	1-2, 3-4, 5-6, 7-8
W11	1-2 Standard I/O Mapping none User-programmed I/O mapping	1-2
W12	Affects **Memory Mapping** and response of onboard memory to MEMEX and Segment. See figure 2-5.	1-2, 5-6, 9-10, 11-12
W13-W22	Affects RS-232-C port. See figure 2-4. W13-W17 are not used by the 7807.	W15-W17 and W20-W22, 1-2, 3-4, W13, W14, W18, W19, No Jumper

Figure 2-2. (Continued)

CARD LAYOUT AND JUMPERS

Several optional functions on the 7806/7 are selected by jumpers. Most of the jumpers are wirewrap posts which can be shorted by shorting plugs or wirewrapping. Additional shorting plugs may be obtained from Elco Corp., part number 00 8261 02 42 00 870. Two of the jumpers are soldered-in wire jumpers. These can be disconnected by cutting them. Be sure to cut them in two places to prevent reshorting. The wire jumpers can be replaced with a pair of wirewrap posts (Elco Corp., part number 00 8261 02 32 00 852). When removing the wire jumpers, be sure to cut them and remove one half at a time to prevent circuit trace damage.

CABLING

Figure 2-3 is a list of cables which can be used with the 7806/7.

7806/7 CONNECTOR	PRO-LOG CABLE	DESCRIPTION
7807 J1 Parallel Port Connector	RC709-1 or RC709-2	2 feet (RC709-1) or 3 feet (RC709-2), terminates in 36-pin female Centronics-type connector with bail lock.
7806 J1 and 7806/7 J2 Serial Port Connector	RC701-3 or RC702-3	3 feet, terminates in 25-pin female (RC701-3) or male (RC702-3) D connector with RS-232-C pinout.
J3 CTC and W/Ready Lines Connector	RC704-1 or RC704-2	2-pin connector (same on both ends) with 1 foot (RC704-1) or 2 feet (RC704-2) of twisted pair cable, can mate to interrupt output connectors of other cards or can mate to 7605 TTL I/O card for Segment.
J4 Segment Connector		

Figure 2-3. Cables for Use with 7806/7

SERIAL CHANNEL

There are five jumper groups affecting the serial channel (figure 2-4). They select whether the serial channel operates as data communications equipment (DCE) or data terminal equipment (DTE). Generally, when operating the channel as DCE, the three main jumpers for that channel (20, 21, and 22 also 15, 16, and 17 for the 7806) should be jumpered across pins 1-2 and 3-4. To operate the channel as DTE, turn the jumpers sideways to short pins 1-3 and 2-4.

If the channel is interfaced to a modem, it may be necessary to monitor the Data Carrier Detect (DCD) and Ring Indicator (RI) lines, whereas the Data Set Ready (DSR) line may not be needed. To provide for this, remove jumper 2-4 of jumper group W20 and W15 for the 7806 and install jumpers at W18 and W19 and also at W14 and W13 for the 7806.

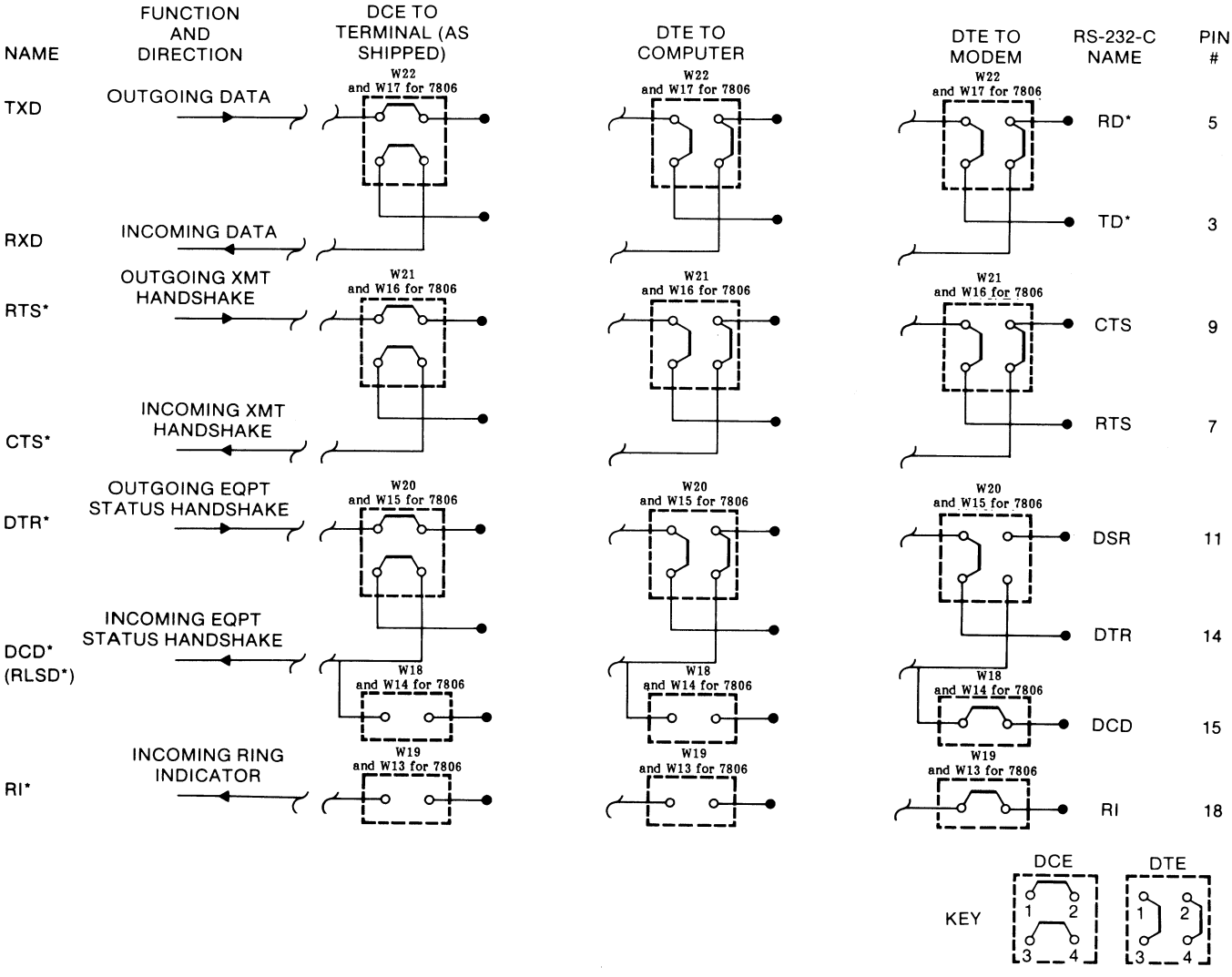


Figure 2-4. Serial Port DCE/DTE Jumpers (7806/7)

MEMORY

The 7806/7 has four sockets which accept memory parts in a variety of sizes, ROM or RAM. Memory parts used must have an access time no greater than 250 nanoseconds for 4 MHz and 3.6864 MHz versions (7806/7-0 and -1) or 450 nanoseconds for 2.5 MHz versions (7806/7-2). There are eight jumpers affecting memory mapping which are explained under the subheadings below. Memory decoding on the 7806/7 is controlled by a bipolar PROM. The PROM contains 16 optional memory mapping schemes, seven of which have been preprogrammed. The rest are blank and may be programmed by the user. The active option is selected by jumper. The signals controlling memory include the STD BUS memory expansion line (MEMEX), which allows two 64K memory banks in a system, and a Segment line. Segment is an external signal decoded by this and most Pro-Log memory cards. It allows a virtually unlimited number of memory banks in a system.

Selecting Memory Mapping

Jumper W12 selects memory mapping (figures 2-2 and 2-5). There are seven preprogrammed options in the memory decoder PROM from which to choose. The rest are blank and may be programmed by the user (see Memory Decoder PROM, page 2-10). MEMEX and Segment also enter into the memory decoding (see MEMEX and Segment, page 2-9). In selecting a memory option it is important that, whether by jumper or by MEMEX or Segment, the right combination of highs and lows must be applied to the address line inputs of the memory decoder PROM.

The memory option selects only the size of memory space, if any, to be occupied by each socket. Whether a socket is enabled or not and whether it is configured for ROM or RAM is determined by the jumpers described in the following headings.

Configuring the Memory Sockets

After a memory option has been selected, the memory sockets must be configured accordingly. They must be configured for the correct size and may be configured for either ROM or RAM. Figure 2-6, page 2-8, shows the jumper arrangement for each memory type. Jumpers W3-W6 are used for this purpose and correspond to sockets 3-0 respectively. Figure 2-2, page 2-2, shows the locations of the sockets and jumpers.

Disabling Unused Sockets

Unused memory sockets can be disabled by removing jumpers from W10. Jumpers 1-2 through 7-8 correspond to sockets 3-0 respectively. Removing a jumper causes the processor card to expect memory to be offcard over the address range of the socket, thus causing the socket to effectively disappear from the memory map. If a socket does not occupy any memory space according to the memory decoder PROM option, the socket does not need to be disabled. Sockets that are disabled should also be left empty or data bus contention may occur.

OPTION #	A8[1]	A7[2]	A6	A5	SOCKET 0	SOCKET 1	SOCKET 2	SOCKET 3
0	LO	LO	LO	LO	4K 0000-0FFF	4K 1000-1FFF	4K 2000-2FFF	2K 3000-37FF
1	LO	LO	LO	HI	8K 0000-1FFF	8K 2000-3FFF	8K 4000-5FFF	8K 6000-7FFF
2	LO	LO	HI	LO	16K 0000-3FFF	16K 4000-7FFF	8K 8000-9FFF	8K A000-BFFF
3	LO	LO	HI	HI	32K 0000-7FFF	8K 8000-9FFF	8K A000-BFFF	8K C000-DFFF
4	LO	HI	LO	LO	4K 0000-0FFF	4K 1000-1FFF	4K 2000-2FFF	4K 3000-3FFF
5	LO	HI	LO	HI	16K 0000-3FFF	12K [3] 4000-6FFF	2K 7000-77FF	2K 7800-7FFF
6	LO	HI	HI	LO	2K 0000-07FF	2K 0800-0FFF	2K 1000-17FF	2K 1800-1FFF
7	LO	HI	HI	HI				
8	HI	LO	LO	LO				
9	HI	LO	LO	HI		UNPROGRAMMED, MAY BE PROGRAMMED BY THE USER		
10	HI	LO	HI	LO				
11	HI	LO	HI	HI				
12	HI	HI	LO	LO				
13	HI	HI	LO	HI				
14	HI	HI	HI	LO				
15	HI	HI	HI	HI				

NOTE: With jumper W12 configured as shipped and with MEMEX low, as it is after power-up or reset, option #0 is selected.
[1]May be connected to MEMEX
[2]May be connected to SEGMENT
[3]Requires a 16K device be installed, but only 12K of the device may be accessed.

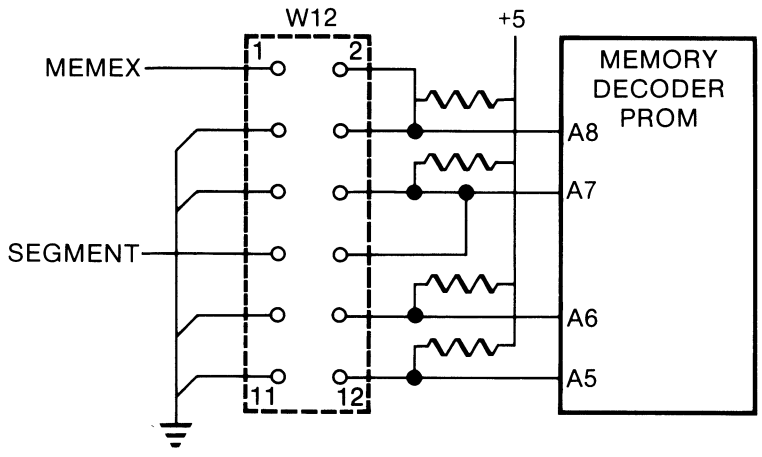
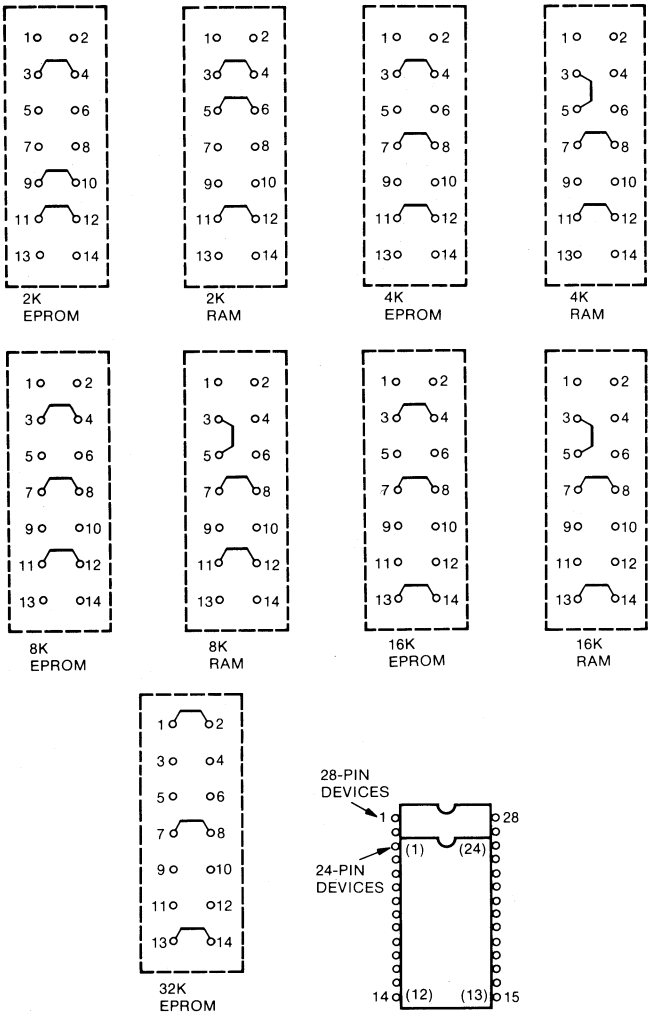


Figure 2-5. Memory Mapping Options (7806/7)



NOTES: 1. Jumper groups W3-W6 correspond to sockets 3-0, respectively.

2. 4K and 16K RAMs do not exist as of this writing; the figure is based on the assumption that if they do become available they will conform to the JEDEC 28-pin standard.

Figure 2-6. Memory Socket Option Jumpers (7806/7)

MEMEX

MEMEX is the STD BUS memory expansion line. It selects one of two banks of memory with up to 64K of memory per bank. It can be used to expand memory or to control a bootstrap PROM. In bootstrapping, the bootstrap PROM is enabled initially. After bootstrapping, the MEMEX line is complemented, disabling the PROM and enabling a block of RAM in its place. The 7806/7 controls MEMEX through an onboard output port.

There are three jumpers affecting MEMEX (figure 2-2, page 2-2). Jumpers W1 and W9 affect the 7806/7 control of MEMEX. Jumper W12 pins 1-2 and the memory decoder PROM affect the response of the onboard memory to MEMEX (figure 2-5, page 2-7).

With jumper W12 pins 1-2 removed, the onboard memory does not respond to MEMEX at all. With the jumper in place, the onboard memory responds to MEMEX according to the way the memory decoder PROM is programmed. Note that jumper W12 pins 3-4 should not be shorted if pins 1-2 are shorted.

When MEMEX changes state, a different memory decoder PROM option is selected. For instance, if option 0 is selected when MEMEX is low, option 8 will be selected when MEMEX is high.

The two banks MEMEX selects may be either one bank onboard and one bank offboard or both banks onboard. If bootstrapping is used, both banks may be onboard. The bootstrap EPROM may be in sockets 0 and 1, and the RAM replacement in sockets 2 and 3.

Segment

The Segment signal is similar to MEMEX. However, it is not an STD BUS signal. Segment would typically be generated by a port on an I/O card and connected to a memory card or the 7806/7 by a cable. Most Pro-Log memory cards decode Segment.

Several Segment lines may be generated by different ports or bits of a port. Only one is active at a time, allowing selection of one of any number of memory banks.

With jumper W12 pins 7-8 removed, onboard memory does not respond to Segment (figures 2-2 and 2-5, pages 2-2 and 2-7). With pins 7-8 installed, Segment selects one of two memory decoder PROM options. For instance, if option 0 is selected with Segment low, option 4 is selected with Segment high. Note that jumper W12 pins 5-6 and 7-8 must not be connected at the same time.

Like MEMEX, the effect of Segment on onboard memory is a function of the memory decoder PROM.

Memory Decoder PROM

The memory decoder PROM is a 512 by 8 bipolar PROM. Figure 2-7 is a list of parts which can be used as the decoder PROM and the Pro-Log PROM programming equipment to program them. The same PROM type and programming equipment are used to program the I/O decoder.

The memory decoder PROM contains 16 memory mapping options. Seven of these have been preprogrammed (figure 2-5, page 2-7); the rest are blank and may be programmed by the user. Each option occupies 32 bytes in the PROM and each byte represents a 2K-byte section of memory. Depending on the data programmed into each location, the decoder selects one of the onboard memory sockets or offcard memory over the 2K address space represented by that byte.

PROM PART #	Pro-Log PROM Programming Equipment*		
	PERSONALITY MODULE	PINOUT ADAPTER	CONFIG-URATOR
MMI 6348-1 Harris 7648-5	PM9037 PM9039A	PA20-1 PA20-1	512x8(H) 512x8(H)

*The equipment listed must be used in conjunction with any Pro-Log Series 90 PROM Programmer.

Figure 2-7. Decoder PROM Part Numbers and Programming Equipment (7806/7)

Figure 2-8 shows the 32-byte section of the PROM representing each option, the address range affected by each byte within an option, and the data byte to be programmed into each byte to select an onboard socket or offcard memory. As an example, figure 2-9, page 2-12, shows the program already in the decoder PROM for option 0. If a socket is to occupy more than 2K of memory space, simply repeat the appropriate data byte as many times as necessary. Note that in the example sockets 0, 1, and 2 each occupy 4K of memory space.

If MEMEX or Segment are to be used, two options need to be programmed: one for when the signal is low and the other when it is high. Refer to figure 2-5, page 2-7, and MEMEX and Segment on the previous page.

I/O MAPPING

Two jumpers affect I/O mapping. I/O mapping is controlled by a bipolar PROM in much the same way as memory mapping. The I/O decoder PROM has one mapping option preprogrammed into it and one option which is blank. The 7806 occupies nine I/O addresses, four for the CTC, four for the DART and one for MEMEX. The 7807 occupies these same nine I/O addresses plus one more for the 8-bit parallel output port. By programming the second option of the I/O decoder PROM, the user may select almost any I/O mapping plan.

DECODER PROM ADDRESS	OPTION #
000-01F	0
020-03F	1
040-05F	2
060-07F	3
080-09F	4
0A0-0BF	5
0C0-0DF	6
0E0-0FF	7
100-11F	8
120-13F	9
140-15F	10
160-17F	11
180-19F	12
1A0-1BF	13
1C0-1DF	14
1E0-1FF	15

DATA BYTE	MEMORY ENABLED
FF	OFF CARD
EE	SOCKET 0
DD	SOCKET 1
BB	SOCKET 2
77	SOCKET 3

BYTE # WITHIN AN OPTION		AFFECTED ADDRESS RANGE
DEC	HEX	
1	00	0000-07FF
2	01	0800-0FFF
3	02	1000-17FF
4	03	1800-1FFF
5	04	2000-27FF
6	05	2800-2FFF
7	06	3000-37FF
8	07	3800-3FFF
9	08	4000-47FF
10	09	4800-4FFF
11	0A	5000-57FF
12	0B	5800-5FFF
13	0C	6000-67FF
14	0D	6800-6FFF
15	0E	7000-77FF
16	0F	7800-7FFF
17	10	8000-87FF
18	11	8800-8FFF
19	12	9000-97FF
20	13	9800-9FFF
21	14	A000-A7FF
22	15	A800-AFFF
23	16	B000-B7FF
24	17	B800-BFFF
25	18	C000-C7FF
26	19	C800-CFFF
27	1A	D000-D7FF
28	1B	D800-DFFF
29	1C	E000-E7FF
30	1D	E800-EFFF
31	1E	F000-F7FF
32	1F	F800-FFFF

Figure 2-8. Memory Decoder PROM Programming Information (7806/7)

PRO-LOG CORPORATION

PROGRAM ASSEMBLY FORM

HEXADECIMAL PAGE ADR	HEXADECIMAL LINE ADR	HEXADECIMAL INSTR.	MNEMONIC LABEL	MNEMONIC INSTR.	MNEMONIC MODIFIER	TITLE MEMORY DECODING DATE / COMMENTS
0	00	EE				ENABLE ROM/RAM 0
0	01	EE				
0	02	DD				ENABLE ROM/RAM 1
0	03	DD				
0	04	BB				ENABLE ROM/RAM 2
0	05	BB				
0	06	77				ENABLE ROM/RAM 3
0	07	FF				EXTERNAL MEMORY
0	08					
0	09					
0	0A					
0	0B					
0	0C					
0	0D					
0	0E					
0	0F					
0	10					
0	11					
0	12					
0	13					
0	14					
0	15					
0	16					
0	17					
0	18					
0	19					
0	1A					
0	1B					
0	1C					
0	1D					
0	1E					
0	1F	FF				

100001 2/77

Figure 2-9. Memory Decoder PROM Example, Option #0 (7806/7)

I/O Decoder

Figure 2-7, page 2-10, is a list of the type of PROMs which can be used as I/O decoder PROMs and the Pro-Log PROM programming equipment to program them. I/O decoder PROM program development differs from that of the memory decoder PROM.

The I/O PROM has 512 bytes. The first 256 represent the first option, and the last 256 represent the second option. Every byte represents one I/O address. Program each location with the data (figure 2-10) to select the desired function for the corresponding I/O address.

The DART and CTC each occupy four I/O addresses and should have four addresses programmed for their use. Since they each decode address lines A0 and A1, they should occupy four consecutive addresses.

The second option, programmed by the user, is activated by removing jumper W11 (see figure 2-2, page 2-2).

DATA BYTE	I/O SELECTED
FF	OFF CARD
F6	CTC
F5	DART
F3	MEMEX
E7	PARALLEL LATCH (7807 ONLY)

Figure 2-10. I/O Decoder PROM Data (7806/7)

IOEXP

IOEXP is the I/O expansion line of the STD BUS. The 7806/7 does not control IOEXP or decode it for its own use. Most Pro-Log I/O cards do decode it and must have it low to operate. IOEXP is connected to ground on the 7806/7. If another card in the system controls IOEXP, jumper W2 should be removed. This leaves IOEXP floating and free to be controlled externally.

PRIORITY CHAIN

The 7806/7 makes use of the STD BUS interrupt priority chain. Priority is based on position in the card rack with the highest priority on the right. (Note that some manufacturers make card racks with highest priority on the left.) The 7806/7 and other cards in a system which make use of the priority chain should be installed in the card rack according to their priority.

The 7806/7 has a fixed priority for the various interrupt sources on the card but, as a card, it may be higher or lower than other cards.

Cards that use the priority chain must be placed next to each other with no unused card slots between them. Other cards which do not use the priority chain may be installed between cards using the priority chain, but these other cards must pass on the priority chain. They must connect pins 51 and 52 (priority chain out and priority chain in) together onboard. This is standard practice on Pro-Log STD BUS cards.

The cards which use the priority chain as a group can be placed anywhere in the card rack, and other cards can be placed anywhere on either side of the group. Other cards may have empty slots between them or may be placed in the group provided they pass on the priority chain.

SPECIFICATIONS

Figures 2-11 through 2-26 on the following pages give specifications for the 7806/7 card.

The timing figures are derived from Z80A CPU timing and timing for onboard circuitry. The typical numbers are timing values for which all the necessary minimum or maximum values were not available.

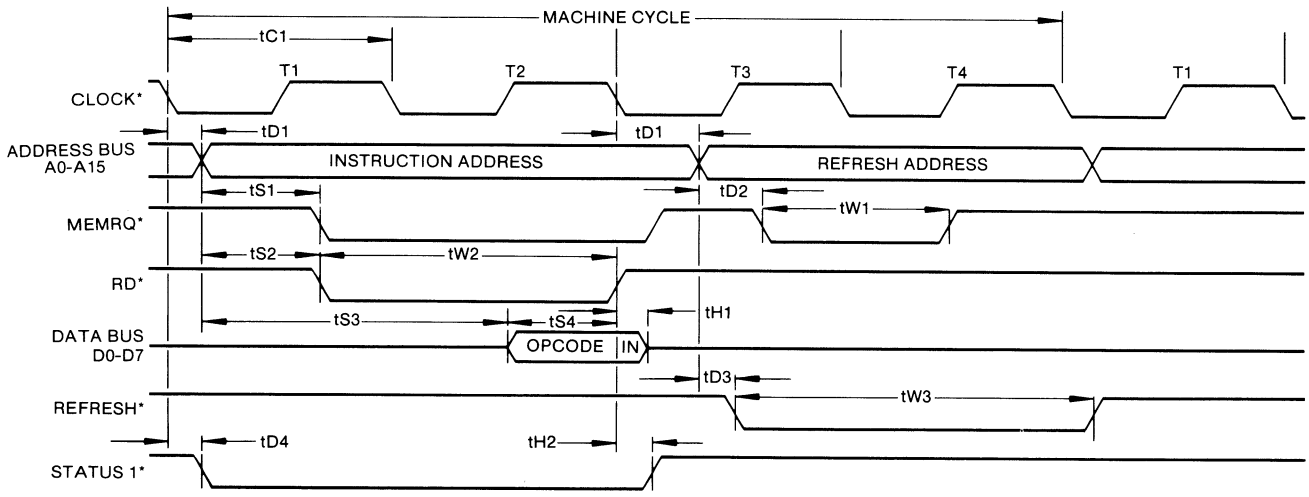
Mechanical Specifications

The 7806/7 meets all general mechanical specifications of the STD BUS.

RESET

Power-on reset on the 7806/7 uses a voltage comparator to be sure Vcc rises to an acceptable operating level before the onboard reset line and the STD BUS SYSRESET* line are released. Once Vcc has reached an acceptable level, a delay of 40 milliseconds maximum occurs before reset is released. This delay gives the onboard clock time to begin functioning.

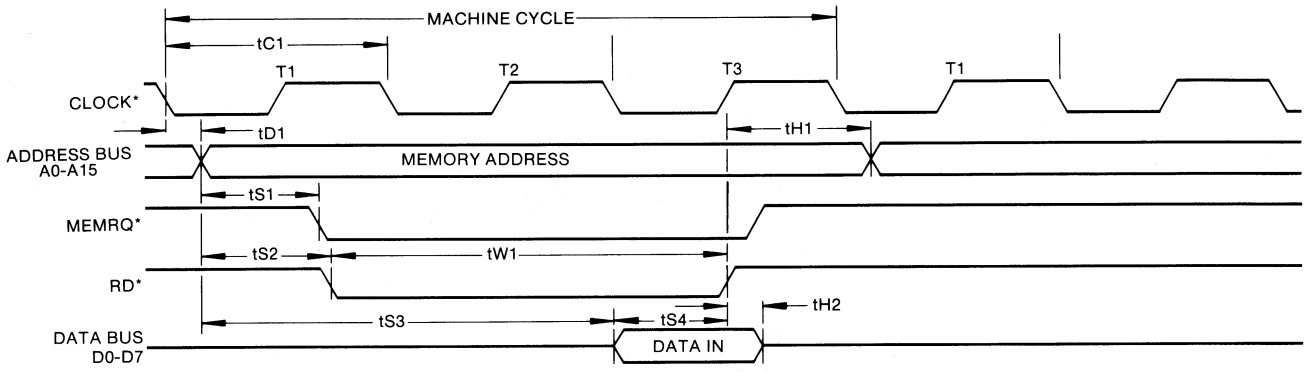
Pushbutton reset is activated by the STD BUS line PBRESET*. The line is debounced approximately 5 milliseconds when it goes low and approximately 20 milliseconds when it goes high. The resulting reset signal is synchronized with the Z80A M1* signal and is 30 microseconds typical to 50 microseconds maximum in duration. This avoids disturbing the refresh cycle of dynamic memories. If the Z80A does not generate an M1* signal, a reset pulse occurs after a delay of approximately 400 nanoseconds.



SYMBOL	DESCRIPTION	4 MHz			2.5 MHz		
		MIN	TYP	MAX	MIN	TYP	MAX
tC1	Clock cycle time	250			400		
tD1	Address delay from CLOCK*			125			160
tD2	MEMRQ* delay from Refresh Address valid		110			150	
tD3	REFRESH* delay from Refresh Address valid		30			45	
tD4	STATUS 1* delay from CLOCK*			105			135
tS1	Address setup before MEMRQ* active	50			110		
tS2	Address setup before RD* active		85			155	
tS3	Address setup before DATA valid	290			540		
tS4	Data setup before RD* inactive	55			70		
tH1	Data hold time after RD* inactive	-5			-5		
tH2	STATUS 1* hold time after RD* inactive		0			15	
tW1	MEMRQ* pulse width (REFRESH)	215			355		
tW2	RD* pulse width	280			465		
tW3	REFRESH* pulse width		485			765	

All times given in nanoseconds.

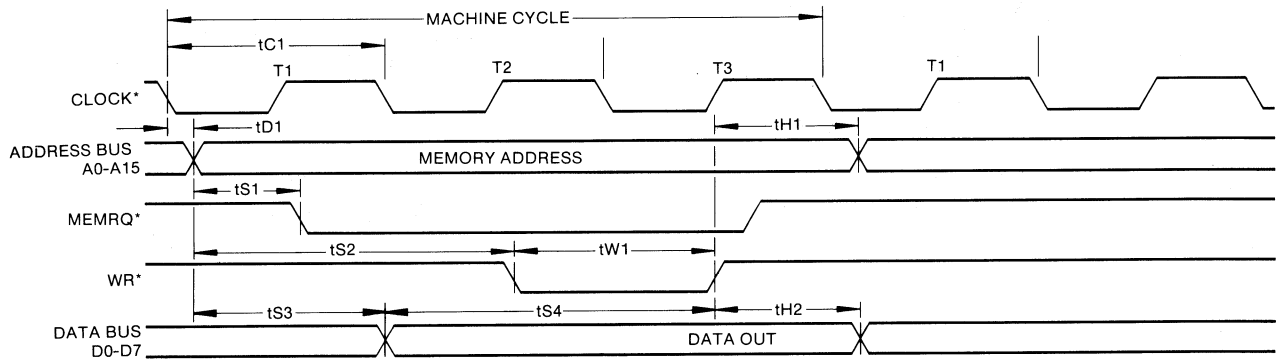
Figure 2-11. Opcode Fetch and Refresh Timing (7806/7)



		4 MHz			2.5 MHz		
SYMBOL	**DESCRIPTION**	**MIN**	**TYP**	**MAX**	**MIN**	**TYP**	**MAX**
tC1	Clock cycle time	250			400		
tD1	Address delay from CLOCK*			125			160
tS1	Address setup before MEMRQ* active	50			110		
tS2	Address setup before RD* active		85			155	
tS3	Address setup before data valid	415			740		
tS4	Data setup before RD* inactive		140			175	
tH1	Address hold time after RD* inactive	50			130		
tH2	Data hold time after RD* inactive	-5			-5		
tW1	RD* pulse width		485			775	

All times given in nanoseconds.

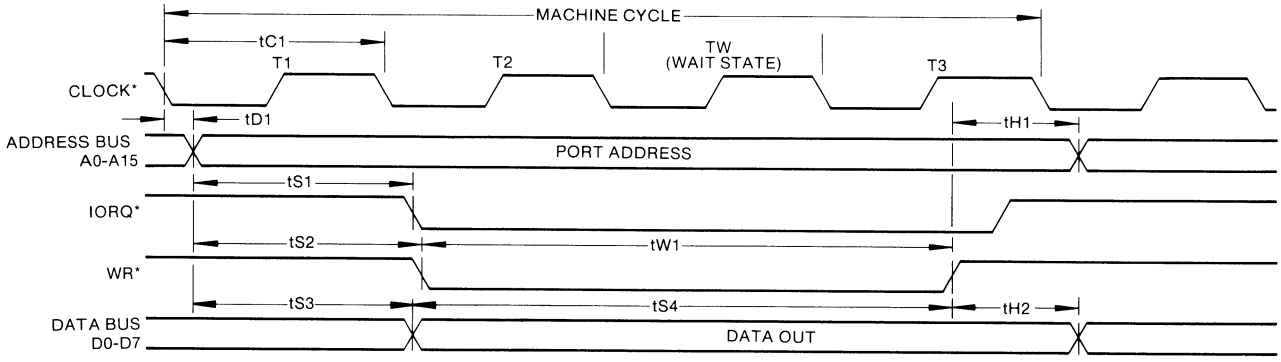
Figure 2-12. Memory Read Timing other than Opcode Fetch (7806/7)



		4 MHz			2.5 MHz		
SYMBOL	**DESCRIPTION**	**MIN**	**TYP**	**MAX**	**MIN**	**TYP**	**MAX**
tC1	Clock cycle time	250			400		
tD1	Address delay from CLOCK*			125			160
tS1	Address setup before MEMRQ* active	50			110		
tS2	Address setup before WR* active		320			515	
tS3	Address setup before data valid		140			255	
tS4	Data setup before WR* inactive	285			535		
tH1	Address hold time after WR* inactive	65			145		
tH2	Data hold time after WR* inactive	30			90		
tW1	WR* pulse width	215			355		

All times given in nanoseconds.

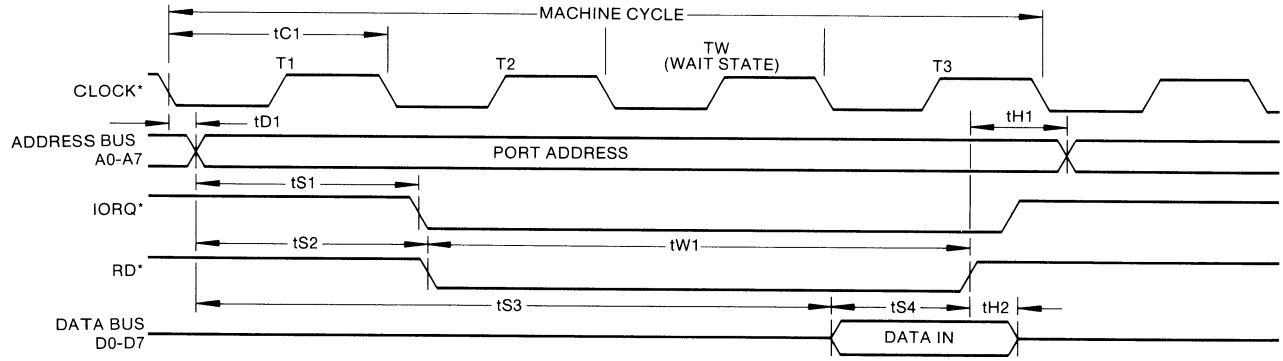
Figure 2-13. Memory Write Timing (7806/7)



		4 MHz			2.5 MHz		
SYMBOL	**DESCRIPTION**	**MIN**	**TYP**	**MAX**	**MIN**	**TYP**	**MAX**
tC1	Clock cycle time	250			400		
tD1	Address delay from CLOCK*			125			160
tS1	Address setup before IORQ* active	165			305		
tS2	Address setup before WR* active		190			320	
tS3	Address setup before data valid		140			255	
tS4	Data setup before WR* inactive		665			1055	
tH1	Address hold time after WR* inactive	65			145		
tH2	Data hold time after WR* inactive	30			90		
tW1	WR* pulse width		625			1000	

All times given in nanoseconds.

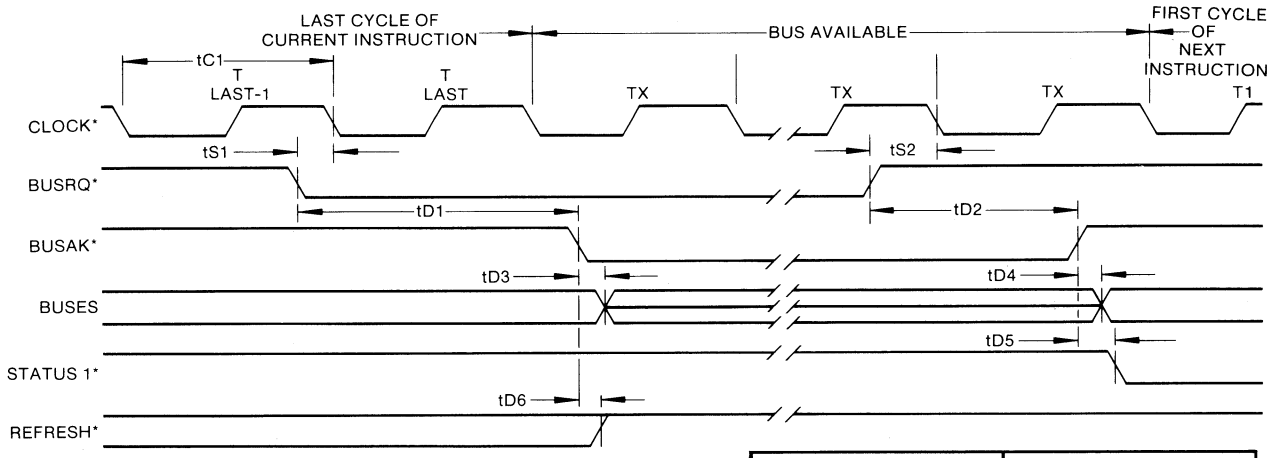
Figure 2-14. Output Timing (7806/7)



		4 MHz			2.5 MHz		
SYMBOL	**DESCRIPTION**	**MIN**	**TYP**	**MAX**	**MIN**	**TYP**	**MAX**
tC1	Clock cycle time	250			400		
tD1	Address delay from CLOCK*			125			160
tS1	Address setup before IORQ* active	165			305		
tS2	Address setup before RD* active		210			340	
tS3	Address setup before data valid	665			1140		
tS4	Data setup before RD* inactive		140			175	
tH1	Address hold time after RD* inactive	50			130		
tH2	Data hold time after RD* inactive	-5			-5		
tW1	RD* pulse width		610			990	

All times given in nanoseconds.

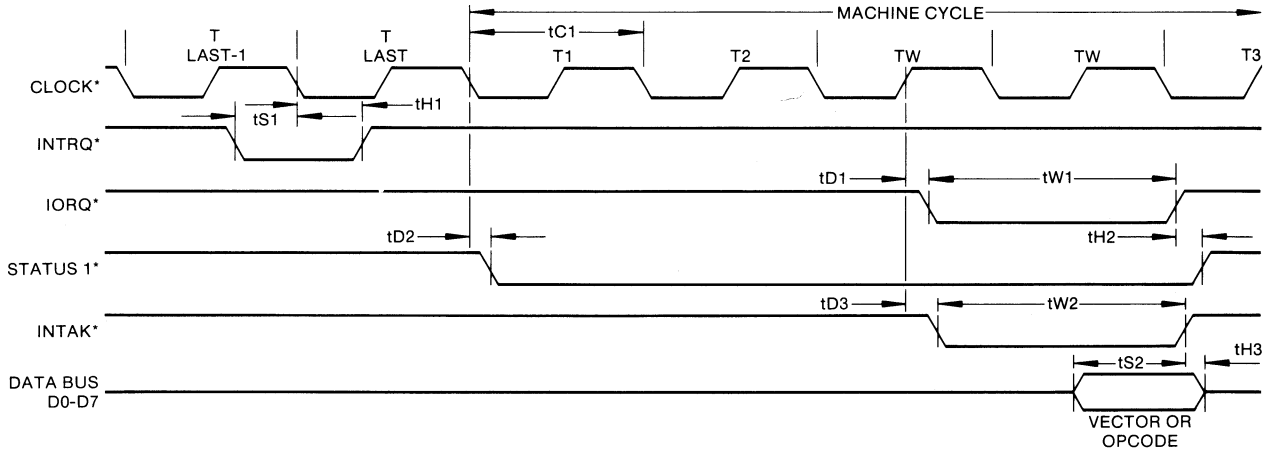
Figure 2-15. Input Timing (7806/7)



SYMBOL	DESCRIPTION	4 MHz MIN	4 MHz TYP	4 MHz MAX	2.5 MHz MIN	2.5 MHz TYP	2.5 MHz MAX
tC1	Clock cycle time	250			400		
tD1	BUSAK* delay from BUSRQ* active		440			640	
tD2	BUSAK* inactive delay from BUSRQ* inactive		335			445	
tD3	Buses delay after BUSAK* active			70			70
tD4	Buses delay after BUSAK* inactive	5			5		
tD5	STATUS 1* delay after BUSAK* inactive		160			250	
tD6	REFRESH* delay after BUSAK* active		30			40	
tS1	BUSRQ* active setup before last time state	70			100		
tS2	BUSRQ* inactive setup before last time state	70			100		

All times given in nanoseconds.

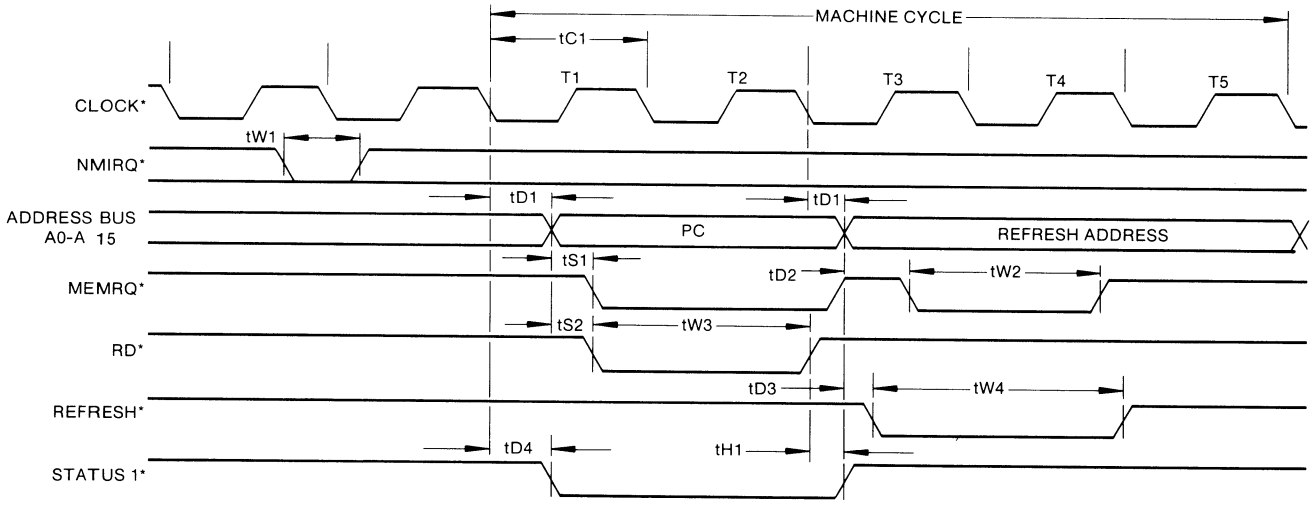
Figure 2-16. Bus Request Timing (7806/7)



SYMBOL	DESCRIPTION	4 MHz MIN	4 MHz TYP	4 MHz MAX	2.5 MHz MIN	2.5 MHz TYP	2.5 MHz MAX
tC1	Clock cycle time	250			400		
tD1	IORQ* delay from CLOCK*			95			120
tD2	STATUS 1* delay from CLOCK*			105			135
tD3	INTAK* delay from CLOCK*			130			155
tS1	INTRQ* setup before last time state	100			100		
tS2	Data setup before INTAK* inactive		135			155	
tH1	INTRQ* hold time after last time state	-5			-5		
tH2	STATUS 1* hold time after IORQ*		0			15	
tH3	Data hold time after INTAK* inactive	-15			-15		
tW1	IORQ* pulse width		360			570	
tW2	INTAK* pulse width		330			540	

All times given in nanoseconds.

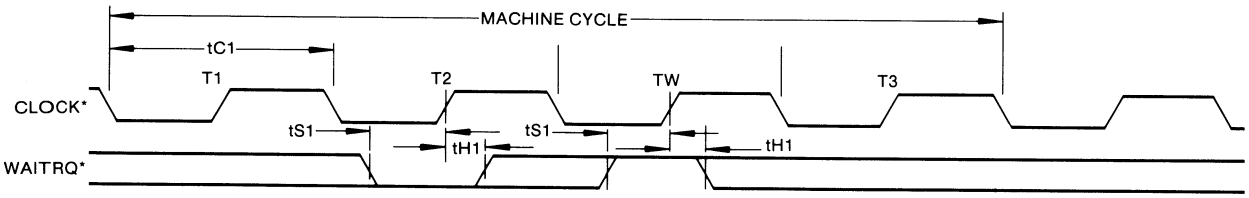
Figure 2-17. Interrupt Request Timing (7806/7)



SYMBOL	DESCRIPTION	4 MHz			2.5 MHz		
		MIN	TYP	MAX	MIN	TYP	MAX
tC1	Clock cycle time	250			400		
tD1	Address delay from CLOCK*			125			160
tD2	MEMRQ* delay from Refresh address valid		110			150	
tD3	REFRESH* delay from Refresh address valid		30			45	
tD4	STATUS 1* delay from CLOCK*			105			135
tS1	Address setup before MEMRQ* active	50			110		
tS2	Address setup before RD* active		85			155	
tH1	STATUS 1* hold time after RD* inactive		0			15	
tW1	NMIRQ* pulse width	80			80		
tW2	MEMRQ* pulse width (REFRESH)	215			355		
tW3	RD* pulse width	280			465		
tW4	REFRESH* pulse width		485			765	

All times given in nanoseconds.

Figure 2-18. Nonmaskable Interrupt Request Timing (7806/7)



SYMBOL	DESCRIPTION	4 MHz			2.5 MHz		
		MIN	TYP	MAX	MIN	TYP	MAX
tC1	Clock cycle time	250			400		
tS1	WAITRQ* setup before CLOCK*	95			95		
tH1	WAITRQ* hold time after CLOCK*	-5			-5		

All times given in nanoseconds.

Figure 2-19. Wait State Timing (7806/7)

PARAMETER		RECOMMENDED OPERATING LIMITS MIN	TYP	MAX	ABSOLUTE NONOPERATING LIMITS MIN	MAX	UNITS
Supply Voltage		4.75	5.00	5.25	0	7.00	V
Supply Current(1)	7806 7807	N/A	1225 1285	1600 1690	N/A	N/A	mA
Positive RS-232-C Supply Voltage(2)		11.40	12.00	12.60	0	15.00	V
Positive RS-232-C Supply Current	7806 7807	N/A	38 19	60 30	N/A	N/A	mA
Negative RS-232-C Supply Voltage(2)		-11.40	-12.00	-12.60	0	-15.00	V
Negative RS-232-C Supply Current	7806 7807	N/A	43 21	65 35	N/A	N/A	mA

NOTES:
(1) Without onboard memory.
(2) The auxiliary voltage source is necessary only if the RS-232-C interface is used.

PARAMETER	RECOMMENDED OPERATING LIMITS MIN	TYP	MAX	ABSOLUTE NONOPERATING LIMITS MIN	MAX	UNITS
Free Air Temperature	0	+25	+65	-40	+75	°C
Noncondensing relative humidity	5	N/A	95	5	95	%

Figure 2-20. Electrical and Environmental Specifications (7806/7)

PARAMETER	MIN	TYP	MAX	UNITS
High-level output voltage(1)	9.0	10.5		Volts
Low-level output voltage(1)	-9.0	-10.5		Volts
High-level short-circuit current	6.0	10.0	12.0	mA
Low-level short-circuit current	-6.0	-10.0	-12.0	mA
High-level input threshold voltage			3.0	Volts
Low-level input threshold voltage			-3.0	Volts

(1)At termination resistance of 3.0K minimum

Figure 2-21. Serial Switching Levels (7806/7)

PIN # OF 7807 PARALLEL CONNECTOR J1	SIGNAL NAME	CENTRONICS PRINTER PIN
1	STROBE*	1
2	GND	19
3	DATA 1	2
4	GND	20
5	DATA 2	3
6	GND	21
7	DATA 3	4
8	GND	22
9	DATA 4	5
10	GND	23
11	DATA 5	6
12	GND	24
13	DATA 6	7
14	GND	25
15	DATA 7	8
16	GND	26
17	DATA 8	9
18	GND	27
19	N/C	10
20	GND	28
21	BUSY	11
22	GND	29
23	PE	12
24	GND	30
25	N/C	13
26	N/C	31

NOTE: Pro-Log cable RC709 can be used to convert from the parallel port connector to a Centronics-type 36-pin connector.

Figure 2-22. 7807 Parallel Connector J1 Pin List

PIN # OF SERIAL CONNECTOR J2 AND 7806 J1	SIGNAL NAME	RS-232-C PIN
1	N/C	1
2	N/C	14
3	TD*	2
4	N/C	15
5	RD*	3
6	N/C	16
7	RTS	4
8	N/C	17
9	CTS	5
10	N/C	18
11	DSR	6
12	N/C	19
13	GND	7
14	DTR	20
15	DCD (RLSD)	8
16	N/C	21
17	N/C	9
18	RI	22
19	N/C	10
20	N/C	23
21	N/C	11
22	N/C	24
23	N/C	12
24	N/C	25
25	N/C	13
26	N/C	N/C

NOTE: Pro-Log cables RC701 and RC702 can be used to convert from the serial port connector to a D connector with the standard RS-232-C pinout (see figure 2-2, page 2-2).

Figure 2-23. Serial Connector Pin List (7806/7)

PIN NUMBER				PIN NUMBER			
OUTPUT (LSTTL DRIVE)				OUTPUT (LSTTL DRIVE)			
INPUT (LSTTL LOADS)				INPUT (LSTTL LOADS)			
MNEMONIC							MNEMONIC
CLOCK/INT 3	5		2	1	OUT		GND
CLOCK/INT 2	5		4	3	OUT		GND
COUNT/TIME 2		10	6	5	OUT		GND
CLOCK/INT 1	5		8	7	OUT		GND
COUNT/TIME 1		10	10	9	OUT		GND
W/RDY A*	1	9	12	11	OUT		GND
W/RDY B*	1	9	14	13	OUT		GND

*Low-level active logic

Figure 2-24. Counter/Timer Connector J3 Pin List (7806/7)

PIN NUMBER				PIN NUMBER			
OUTPUT (LSTTL DRIVE)						OUTPUT (LSTTL DRIVE)	
INPUT (LSTTL LOADS)						INPUT (LSTTL LOADS)	
MNEMONIC							MNEMONIC
SEGMENT	7		2	1	OUT		GND

Figure 2-25. Segment Connector J4 Pin List (7806/7)

PIN NUMBER				PIN NUMBER			
OUTPUT (LSTTL DRIVE)						OUTPUT (LSTTL DRIVE)	
INPUT (LSTTL LOADS)						INPUT (LSTTL LOADS)	
MNEMONIC							MNEMONIC
+5V	IN		2	1		IN	+5V
GROUND	IN		4	3		IN	GROUND
-5V			6	5			-5V
D7	1	55	8	7	55	1	D3
D6	1	55	10	9	55	1	D2
D5	1	55	12	11	55	1	D1
D4	1	55	14	13	55	1	D0
A15	1	55	16	15	55	1	A7
A14	1	55	18	17	55	1	A6
A13	1	55	20	19	55	1	A5
A12	1	55	22	21	55	1	A4
A11	1	55	24	23	55	1	A3
A10	1	55	26	25	55	1	A2
A9	1	55	28	27	55	1	A1
A8	1	55	30	29	55	1	A0
RD*	1	55	32	31	55	1	WR*
MEMRQ*	1	55	34	33	55		IORQ*
MEMEX		50	36	35	OUT		IOEXP (GROUND)
MCSYNC*		55	38	37	55		REFRESH*
STATUS 0*			40	39	55		STATUS 1*
BUSRQ*	5		42	41	55		BUSAK*
INTRQ*	5		44	43	55		INTAK*
NMIRQ*	5		46	45		5	WAITRQ*
PBRESET*	(1)		48	47	55		SYSRESET*
CNTRL*			50	49	55		CLOCK*
PCI	5		52	51	55		PCO
AUX GND	IN		54	53		IN	AUX GND
AUX -V	IN		56	55		IN	AUX +V

*Low-level active logic
(1) Switch input

Figure 2-26. STD BUS Connector Pin List (7806/7)

INTRODUCTION

This section covers the operation of the 7806 and 7807 serial ports, the 7807 parallel port, the counter/timer, and memory expansion line. All of these functions are controlled through I/O ports. Figure 3-1 shows the functions of the ports on the 7806/7.

MEMORY EXPANSION

Two lines on the 7806/7 can be used for memory expansion: the STD BUS MEMEX line and the external signal Segment. The effect of these lines on 7806/7 memory is explained under Memory, page 2-6.

The Segment line is an input and cannot be controlled by the 7806/7. The MEMEX line is controlled by bit 0 of the MEMEX control port (figure 3-1). Port configuration as inverting or noninverting is controlled by jumper W9 (figure 2-2, page 2-2).

I/O PORT ADDR	OUTPUT FUNCTION	INPUT FUNCTION
F0	CTC CHANNEL 0 Control, Vector, and Load Registers	CTC CHANNEL 0 Current Count Register
F1	CTC CHANNEL 1 Control and Load Registers	CTC CHANNEL 1 Current Count Register
F2	CTC CHANNEL 2 Control and Load Registers	CTC CHANNEL 2 Current Count Register
F3	CTC CHANNEL 3 Control and Load Registers	CTC CHANNEL 3 Current Count Register
F4	DART CHANNEL A XMT Data	DART CHANNEL A RCV Data
F5	DART CHANNEL A Control Registers (Parallel Handshake)	DART CHANNEL A Status Registers (Parallel Handshake)
F6	DART CHANNEL B XMT Data	DART CHANNEL B RCV Data
F7	DART CHANNEL B Control Registers	DART CHANNEL B Status Registers
FD (7807 only)	8-BIT PARALLEL OUTPUT PORT	Not Used
FE	MEMEX CONTROL	Not Used

Figure 3-1. I/O Port Functions (7806/7)

SERIAL CHANNEL AND PARALLEL HANDSHAKE

DART Internal Structure

Operating the 7806 and 7807 serial ports or 7807 parallel handshake lines is mainly a matter of using the control chip (DART) internal registers. Figure 3-2 shows the structure of the internal registers of one channel of the dual channel asynchronous receiver/transmitter (DART). The structure of the two channels is identical. Channel A has been used to implement the 7807 parallel port handshake lines and one of the 7806 serial ports; Channel B has been used to implement the serial RS-232-C port.

Each channel has six control registers, a Transmit (XMT) Data register, three status registers, and a Receive (RCV) Data register. Each channel occupies two I/O addresses with registers Control 0 and Status 0 at one address and XMT and RCV Data at the other. Through these ports, direct access can be gained to Control 0, Status 0, and XMT and RCV Data registers. The other five control registers and two status registers are addressed through the Control 0 register.

The lower three bits of Control 0 are called the register pointer. To access a register through Control 0, a data byte is written to Control 0 with the lower three bits set equal to the number of the register to be accessed. The next time the control/status port address is written to or read from, the designated register is accessed. Control 0, with its register pointer, must be rewritten each time a register is to be accessed through Control 0.

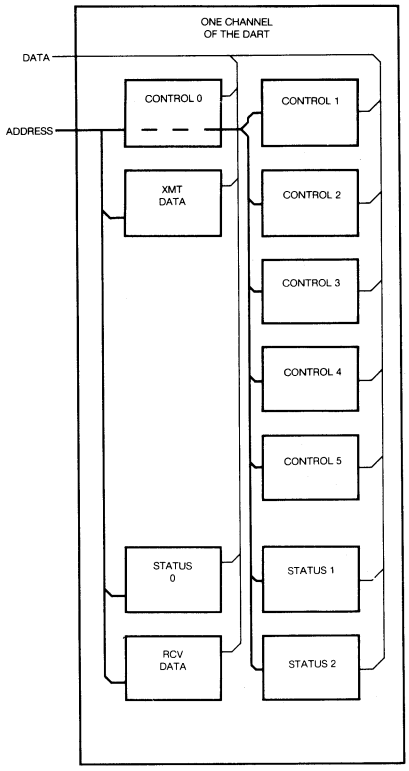


Figure 3-2. DART Internal Structure (7806/7)

Explanation of Registers

Figure 3-3 gives the basic functions of each register. Figure 3-4 and the accompanying notes give a more complete description. After reading the detailed descriptions which follow and becoming familiar with the registers, figures 3-3 and 3-4 should be sufficient for reference. Figure 3-3 can help you find the desired function at a glance and figure 3-4 gives the details. These and other useful figures are repeated in the Quick Reference section at the back of this manual.

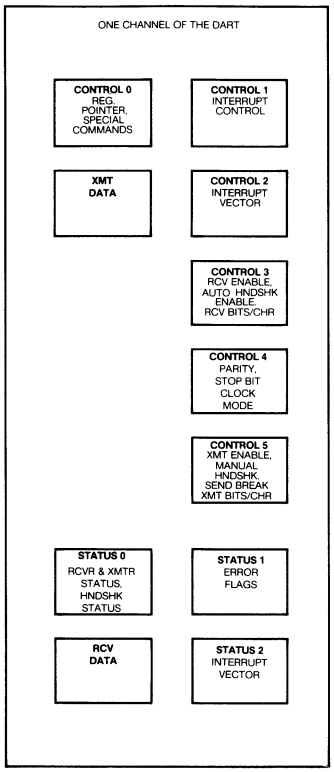


Figure 3-3. DART Register Functions, General (7806/7)

REG	BIT 7	6	5	4	3	2	1	0
RC0			SPECIAL COMMANDS[1]			REGISTER POINTER		
RC1	W/Ready Enable	W/Ready acts as Ready	W/Ready on RCV	RCVR INT Control[2]		Status Affects Vector †	XMTR INT Enable	Hndshk INT Enable
RC2	INTERRUPT VECTOR †							
RC3	RCVR Bits/ Character[3]		Auto Handshake					RCVR Enable
RC4	Clock Divider[4]				Stop Bits[5]		Parity Even	Parity Enable
RC5	DTR/DSR STROBE* (7807 CH A)	XMTR Bits/Character[3]		Send Break	XMTR Enable		RTS/CTS	
RS0	Break		CTS/RTS PE (7807 CH A)	RI	DCD/DSR/DTR BUSY (7807 CH A)	XMTR Buffer Empty	INT Pending §	RCV Char. Available
RS1		Framing Error	RCVR Overrun	Parity Error				Thru Sending
RS2	INTERRUPT VECTOR[6] †							

† Channel B only; § Channel A only

[1] 000 NOP
001 Invalid
010 Reset handshake interrupt
011 Channel reset
100 Re-enable INT on first RCVD character
101 Reset pending XMTR INT
110 Error reset
111 Return from INT*

[2] 00 RCVR INT disabled
01 RCVR INT on first character
10 RCVR INT on every character, parity affects vector
11 RCVR INT on every character, parity does not affect vector

[3] 00 5 Bits/character or less.
01 7 Bits/character
10 6 Bits/character
11 8 Bits/character

When transmitting 5 or less bits/character the following pattern must be followed:

D7	D6	D5	D4	D3	D2	D1	D0	
1	1	1	1	0	0	0	D	One Bit
1	1	1	0	0	0	D	D	Two Bits
1	1	0	0	0	D	D	D	Three Bits
1	0	0	0	D	D	D	D	Four Bits
0	0	0	D	D	D	D	D	Five Bits

D=DATA

[4] 00 Clock divided by 1
01 Clock divided by 16
10 Clock divided by 32
11 Clock divided by 64

[5] 00 Invalid
01 1 Stop Bits
10 1½ Stop Bits
11 2 Stop Bits

[6]* If the Status Affects Vector mode is enabled (RC1), bits 3, 2, and 1 of the interrupt vector will be:

	* B3	B2	B1	
	0	0	0	XMTR Buffer Empty
CH	0	0	1	Handshake Change
B	0	1	0	RCVD character ready
	0	1	1	Special RCV condition
	1	0	0	XMTR Buffer Empty
CH	1	0	1	Handshake Change
A	1	1	0	RCVD character ready
	1	1	1	Special RCV condition

*Channel B only; **Channel A only

Figure 3-4. DART Register Functions, Detailed (7806/7)

RCV Data Register

The RCV Data register is directly addressable and holds the data received over the RS-232-C interface. It holds up to three bytes in addition to the byte currently being received. The error flags in the Status 1 register are also four levels deep so the appropriate status information is available with each data byte. Status 1 should be read before the RCV Data register as the stacks for both data and status are incremented when the data is read. If enabled, an interrupt occurs each time a character reaches the top of the stack. If there are no characters pending in the stack, a received character goes directly to the top of the stack.

If four complete bytes have been received since the last time the processor read a byte out of the RCV Data register, the third byte will be overwritten. The fourth character, in the shift register, is loaded into the bottom of the stack of received characters as soon as it is completely received. If the Status Affects Vector mode is selected (discussed in the DART Operation section, page 3-11), an interrupt is generated with a special vector indicating an error. This interrupt occurs when the character which overwrote the previous one reaches the top of the stack. An error flag is set in the status information associated with that fourth byte, indicating that it overwrote the previous byte.

In the Interrupt on First Received Character mode (interrupts can occur on the first character or every character), if an error occurs the data and status stacks are frozen, even if read, and will prevent access to further data until an error reset is performed. Error reset is explained in the Control 0 description following.

For troubleshooting, it may be useful to note that a received character may not be available to be read for up to 13 system clock cycles, counting from the clock cycle corresponding to the sampling of the last bit of the character.

XMT Data Register

The XMT Data register is directly addressable and holds the data to be transmitted. In addition to the XMT Data register, there is a shift register for the byte currently being transmitted. This allows the processor to give the DART one byte while it is transmitting another, thereby eliminating delays between characters.

An interrupt and/or flag indicates when the XMT Data register is empty and is able to accept another character. The Thru Sending flag in Status 1 indicates that actual transmission has been completed. The Thru Sending condition does not generate an interrupt.

Register Control 0

The Control 0 register is directly addressable and serves two functions. First, its lower three bits act as the register pointer, which allows access to Control registers 1-5 and Status registers 1 and 2. Its second function is to accept special commands. These special commands are given in note one of figure 3-4. A description of each command follows.

NOP: This is a No Operation command which can be issued while setting the register pointer when no other effect is desired.

Invalid: This is an illegal command code and would have an undetermined effect.

Reset Handshake Interrupt: If an incoming handshake line changes state, or a break or end of break is detected on the incoming data line, bits 3, 4, 5, and 7 of the Status 0 register become frozen and an interrupt is generated if enabled (Control 1, bit 0). This allows momentary changes to be noted by the processor. The Reset Handshake Interrupt command releases Status 0. While these bits are frozen, no other changes can be detected and no interrupts are generated by these bits. When the register is released, if a handshake line has changed or a break has started or stopped, the register is updated and then again freezes bits 3, 4, 5, and 7. An interrupt can also be generated. If no such change has occurred, the Handshake and Break flags simply indicate the current status. Therefore, to be sure to get the current status when reading bits 3, 4, 5, or 7 of Status 0, issue a Reset Handshake Interrupt command immediately prior to reading Status 0. The other flags in the register are always free to change according to current conditions.

Channel Reset: This command performs the same function as a system reset, but on only one channel of the DART. On Channel A it also resets the DART prioritization logic. All Control registers must be rewritten for the affected channel. Four clock states (1 microsecond) should be allowed to pass before that channel is written to again. One Z80A NOP instruction, or any other instruction, is sufficient for this delay.

Re-Enable Interrupt on First Received Character: If the Interrupt on First Received Character Only mode is selected (Control 1, bits 3 and 4), it must be re-enabled between received messages. This command performs that function. One application for this mode is in block transfers, which are discussed in the DART Operation section, page 3-11.

Disable Transmitter Interrupt: After the last character in a transmission has been sent and the DART generates an interrupt requesting another character (assuming the Transmitter Interrupt is enabled), this command turns off the interrupt. After the next time a character is sent and the DART is ready for a character, the Interrupt (Transmit vector) again occurs.

Error Reset: The Parity Error and Overrun flags in Status 1 are latching. This allows errors occurring in block transfers to be examined only at the end of the block (block transfers are exlained in the DART Operation section, page 3-11. This command clears the flags.

Return from Interrupt: This command has the same effect as a return-from-interrupt command performed by the processor (see DART Operation, page 3-11). This command can only be issued to Channel A, but affects the whole DART by resetting the interrupt under service latch of the DART device under service, allowing lower priority devices to interrupt.

Register Control 1

This register is addressed through Control 0 and contains the interrupt control bits.

Hndshk Int. Enable (B0): Enables interrupts to occur as a result of changes on the handshake lines or the beginning or ending of a break condition. Any unused handshake lines should be pulled up if handshake interrupts are enabled. The handshake pins at the DART are DCD*, RI*, and CTS*. The signals these pins are connected to are BUSY (Channel A, 7807), DSR/DTR or RLSD (Channel A, 7806 depending on W13 and W14, and Channel B, depending on W18 and W20); N/C (Channel A, 7807), RI (Channel A, 7806, and Channel B); PE (Channel A, 7807), CTS/RTS (Channel A, 7806 depending on W16, and Channel B, depending on W21).

XMT Int. Enable (B1): Enables interrupts (Transmit vector) when the XMT Data register is empty.

Status Affects Vector (B2): If enabled, provides different vectors for different interrupt conditions (see note 6, figure 3-4, page 3-4). This bit can be set only in channel B but affects the whole DART.

RCVR Int. Control (B3, B4): Selects Receiver Interrupt mode (see note 2, figure 3-4, page 3-4). In the Parity Affects Vector mode if a parity error occurs, the Special Receive Condition vector will be returned instead of the Received Character Ready vector. The Status Affects Vector bit must also be set.

W/Ready on RCV (B5): Selects the line for use in the Receive (1) or Transmit (0) mode. The W/RDY A/B* (Wait/Ready) line is used in block transfers (refer to DART Operation, page 3-11).

W/Ready Acts as Ready (B6): Selects whether the W/Rdy A/B* line acts as Wait (0) or Ready (1). The default condition is Wait. As a Wait line, it is generally used with Z80 block transfer instructions to synchronize the CPU to the DART data rate. The Wait line becomes active whenever the processor attempts to read or write data before the DART has data in its receiver or can accept data in its transmitter. The Wait line is open drain and occurs just after the negative edge of the system clock. When using the Wait line, take care not to disturb dynamic RAM refresh.

As a Ready line, it is usually used with a DMA controller. It is active whenever the receiver has a character or whenever the transmitter is empty. The Ready line is actively driven and occurs just after the positive edge of the system clock.

W/Ready Enable (B7): The W/RDY A/B* line is high (in the Ready mode) or floating (in the Wait mode) until enabled by this bit.

Register Control 2

Control 2 is accessed through Control 0 and is used to program the Interrupt vector. This register can only be set in Channel B but affects the whole DART. The vector is returned exactly as sent unless the Status Affects Vector mode is selected. In this case, the vector provided by the DART on the data bus and in Status 2 is affected as shown in note 6 of figure 3-4. The base vector number programmed in Control 2 remains unchanged.

Register Control 3

Control 3 is accessed through Control 0 and contains receiver control bits and the auto handshake bit.

RCVR Enable (B0): This bit enables the receiver and is normally left active throughout operation. If the receiver is disabled while receiving a character, the character is lost and no interrupt occurs.

Auto Handshake (B5): This bit enables the automatic handshake capability of the two incoming handshake lines connected to DART inputs CTS* and DCD* (figure 2-4, page 2-5). If enabled, the DCD* pin, when in the active state, enables the receiver; and the CTS* pin, when in the active state, enables the transmitter. If automatic handshaking is not enabled, these inputs have no effect but may be monitored through Status 0. Transitions on these lines can also cause interrupts if handshake interrupts are enabled (Control 1, bit 0).

RCVR Bits/Character (B6, B7): These determine the number of bits the receiver expects to receive per character (see note 3, figure 3-4, page 3-4). Five is the minimum number. If less than eight bits are used, the character will begin with the least significant bit. The state of the remaining most significant bit(s) is unpredictable and they should be masked off by the processor. An additional parity bit will be received if parity is enabled at register control 4.

Register Control 4

This register is addressed through Control 0 and contains the baud rate clock divider and parity and stop bit control.

Parity Enable (B0): This bit adds a parity bit to the character format and affects both transmitter and receiver.

Parity Even (B1): This bit selects even parity if set and odd parity if cleared. It also affects transmitter and receiver.

Stop Bits (B2, B3): These bits select the number of stop bits transmitted (see note 5, figure 3-4, page 3-4) after each character. The receiver simply checks to be sure there is at least one stop bit.

Clock Divider (B6, B7): These bits select the number by which the baud rate clock is divided (see note 4, figure 3-4). The baud rate clock divided by this number equals the baud rate. The baud rate for the transmitter and receiver of a channel are the same. Since baud rate is a function of the CTC as well, baud rate selection is covered in DART Operation, page 3-11.

The Divide by One mode is not recommended for receiving as it requires external bit synchronization. It can be used for transmitting. In any mode, the DART system clock must be at least five times the baud rate.

Register Control 5

Control 5 is addressed through Control 0 and contains the transmitter and handshake control. All references given below to the state of a pin are as seen at the DART. There is an inversion, however, between the DART and the serial port connector. In most systems, the equipment at the other end of the serial port will also have an inversion.

RTS/CTS (B1): This bit controls the RTS* output pin of the DART. When set, the RTS* pin goes active (low). When reset (to 0), the RTS* pin goes inactive (high) only after the transmitter is empty. If this bit is set inactive (high) while a character is being transmitted, serial data should be retransmitted.

XMT Enable (B3): This bit enables the transmitter. Data is not transmitted and the transmit data pin remains high until the transmitter is enabled. If the transmitter is disabled while sending, the character being sent is completed. An interrupt still occurs when the XMT Data register becomes empty. Only the character being sent will be completed. If one is pending in the XMT Data register, it remains pending.

Send Break (B4): This bit forces the transmit data pin low. Data being transmitted continues to be sent but does not appear on the data line. Thus, characters sent during the break condition are lost. When this bit is reset, the data line is immediately released.

XMT Bits/Character (B5, B6): These bits determine the number of bits per character to be sent by the transmitter. To send less than five, follow the pattern given in note 3 of figure 3-4, page 3-4. Data is transmitted least significant bit first.

DTR/DSR (B7): This bit controls the DART DTR* pin (see figure 2-4, page 2-5). When set, the DTR* pin is active (low). When reset (0), the DTR* pin is inactive (high). This bit provides the STROBE* handshake line for the 7807 parallel port when channel A is selected.

Register Status 0

Status 0 is directly accessible and contains Handshake and DART Status flags. All references given below to the state of a pin are as seen at the DART. The state that appears at the pin is the state that will be seen in the status bit. There is an inversion, however, between the DART and the serial port connector. In most systems, the equipment at the other end of the serial port will also have an inversion.

RCV Char. Avlbl. (B0): This bit indicates a received character is available in the RCV Data register (see RCV Data Register, page 3-5). This bit can generate an interrupt. The bit is reset automatically when the receive buffers are all empty.

Int. Pending (B1): This bit is active if any interrupt is pending for either channel of the DART. It is available only in Channel A. In Channel B, it is always 0.

XMT Buffer Empty (B2): This bit is set whenever the XMT Data register is empty. This bit can generate an interrupt (Transmit vector).

DCD/DSR/DTR (B3): This bit shows the state of the DCD* pin of the DART (figure 2-4, page 2-5). This bit is latched under some circumstances. Refer to the Reset Handshake Interrupts command of register Control 0. A change in this bit can cause an interrupt (Handshake vector). This bit provides the 7807 BUSY handshake line for the parallel port when Channel A is selected on the 7807.

RI (B4): This bit shows the state of the RI* pin of the DART (figure 2-4, page 2-5). This bit can be latched. Refer to the Reset Handshake Interrupt command of register Control 0. A change in this bit can cause an interrupt (Handshake vector).

CTS/RTS (B5): This bit shows the state of the CTS* pin of the DART (figure 2-4, page 2-5). This bit can be latched. Refer to the Reset Handshake Interrupt command of register Control 0. A change in this bit can cause an interrupt (Handshake vector). This bit provides the 7807 PE handshake line for the parallel port when Channel A is selected on the 7807.

Break (B7): This bit indicates that a break condition has been detected on the receive data pin. A break is considered to be a character (dependent on the bits/character preselected) of all zeros plus a 0 Parity bit (if used), plus no Stop bit (logic 0). When this condition is detected, both a Framing Error flag and a Break flag will be set and an interrupt can occur.

If a break begins in the middle of a character, the receiver considers that character to have a framing error and the next character to be a break (if the break continues).

An end of break is detected when the receive data pin returns high. An end of break can also cause an interrupt. This bit can be latched. Refer to the Reset Handhsake Interrupt command of register Control 0.

Register Status 1

This register is accessed through Control 0 and contains data error and transmitter flags.

Thru Sending (B0): This bit indicates that the transmitter is completely through sending.

Parity Error (B4): This bit becomes active when there is a parity error in the character currently available in the RCV Data register (refer to RCV Data Register, page 3-5). This bit then latches and remains latched until the Error Reset command is issued to Control 0. A parity error can generate an interrupt (Special Receive Vector).

RCVR Overrun (B5): This bit becomes active when the character currently available in the RCV Data register has overrun a previous character (refer to RCV Data Register, page 3-5). This bit is then latched until the Error Reset command is issued to Control 0. This bit can cause an interrupt (Special Receive vector).

Framing Error (B6): This bit indicates that a framing error was detected in receiving the character currently available in the RCV Data register (refer to RCV Data Register, page 3-5). After detecting a framing error, the receiver pauses one-half bit time before searching for the next start bit to avoid mistaking the current bit for a start bit.

Register Status 2

This register is accessed through Control 0. It can only be read in Channel B but is indicative of the whole DART. This register allows the Interrupt vector to be read. If an interrupt is pending and the Status Affects Vector mode has been selected, the register shows the modified vector as the processor would see it (see note 6, figure 3-4, page 3-4). At all other times, it shows the vector which was written into Control 2.

DART OPERATION

Initialization

The DART should be initialized in the following order: Write to the counter/timer (CTC) to select a baud rate clock (Channel B only for the 7807), then initialize DART registers 2, 4, 5, 3, and 1 in that order.

Serial Port Baud Rate

Selection of baud rate is a function of both the counter/timer and the DART. Figure 3-5, page 3-12, gives the values to be programmed into the counter/timer and the DART for all possible standard baud rates. To initialize the counter/timer, see page Counter/Timer, page 3-16. Also, software for initializing the DART and for selecting a baud rate is provided in section 4 of this manual.

The 7806/7 is available in three clock speeds: 4 MHz (7806/7-0), 3.6864 MHz (7806/7-1), and 2.5 MHz (7806/7-2). While slower, the 3.6864 MHz and 2.5 MHz clocks divide evenly into standard baud rates allowing two higher baud rates to be used: 19.2K and 38.4K. The 4 MHz clock is intended for high speed processor operation and cannot achieve the highest baud rates. The closest the 4 MHz clock can come to 19.2K baud is 17.8K baud, which is not close enough for consistent operation. The 3.6864 MHz clock can achieve the higher baud rates because the frequency divides into standard baud rates but has a reduced processing speed (approximately 8% slower than the 4 MHz clock). The 2.5 MHz version is for use with slower memories.

General Operation

General operation is largely dependent on the specific application and is mostly a function of the use of the registers. The best way to prepare an operating procedure is to review the functions of the registers with your application in mind. Several aspects of operation are covered in the following headings. They are interrupts, handshaking, and block transfers.

BAUD RATE	CTC CLOCK DIVIDER	CTC COUNT		CTC MODE	DART CLOCK DIVIDER	ACTUAL BAUD RATE % ERROR
		DEC	HEX			
4 MHz (7806/7)						
9600	1	13	0D		16	+0.16
4800	1	26	1A		16	+0.16
2400	1	52	34		16	+0.16
1800	1	69	45		16	+0.64
1200	1	104	68	COUNTER	16	+0.16
600	1	208	D0		16	+0.16
300	16	52	34		16	+0.16
200	16	78	4E		16	+0.16
150	16	104	68		16	+0.16
134.5	16	116	74	TIMER	16	+0.15
110	16	142	8E		16	+0.03
75	16	208	D0		16	+0.16
50	16	78	4E		64	+0.16
3.6864 MHz (7806/7)						
38.4K	1	3	03		16	0
19.2K	1	6	06		16	0
9600	1	12	0C		16	0
4800	1	24	18		16	0
2400	1	48	30	COUNTER	16	0
1800	1	64	40		16	0
1200	1	96	60		16	0
600	1	192	C0		16	0
300	16	48	30		16	0
200	16	72	48		16	0
150	16	96	60		16	0
134.5	16	107	6B	TIMER	16	+0.06
110	16	131	83		16	-0.07
75	16	192	C0		16	0
50	16	72	48		64	0
2.5 MHz (7806/7)						
38.4K	1	2	2		16	+1.72
19.2K	1	4	4		16	+1.72
9600	1	8	8		16	+1.72
4800	1	10	16		16	+1.72
2400	1	21	33	COUNTER	16	-1.36
1800	1	2B	43		16	+0.94
1200	1	41	65		16	+0.16
600	1	82	130		16	+0.16
300	16	21	33		16	-1.36
200	16	31	49		16	-0.35
150	16	41	65		16	+0.16
134.5	16	49	73	TIMER	16	-0.54
110	16	59	89		16	-0.25
75	16	82	130		16	+0.16
50	16	C3	195		16	+0.16

NOTE: Counter mode: Baud rate = ((System clock ÷ 2) ÷ CTC count) ÷ DART clock divider
Timer mode: Baud rate = ((System clock ÷ CTC clock divider) ÷ CTC count) ÷ DART clock divider

Figure 3-5. Baud Rate Selection (7806/7)

Interrupts

The DART has an internal interrupt priority chain. The priorities within the DART and their relationship to CTC interrupts are shown in figure 3-6. The most common way to operate the DART is to use the Status Affects Vector feature and program the Z80 for interrupt mode 2. This is the mode in which the vector is combined with the contents of the Z80 interrupt register to form a 16-bit vector. This vector then points to a location in memory where the address of the interrupt service routine resides. When using this mode, be sure to:

1. Program the Z80 interrupt register. It forms the upper half of the 16-bit vector.

2. Set the least significant bit of the vector programmed into the DART to 0. This is a requirement of the Z80.

3. Use the Z80 Return From Interrupt command at the end of the interrupt service routine. This is a requirement of the DART.

This interrupt plan must be used if the CTC will also be generating interrupts. The CTC can only operate in mode 2.

PRIORITY	DESCRIPTION	VECTOR
Highest		
1	CTC Channel 0	XXXX X000
2	CTC Channel 1	XXXX X010
3	CTC Channel 2	XXXX X100
4	CTC Channel 3	XXXX X110
5	DART Ch. A, RCV character or special RCV condition	XXXX 110X XXXX 111X
6	XMT Buffer Empty	XXXX 100X
7	Handshake Transition	XXXX 101X
8	DART Ch. B, RCV character or special RCV condition	XXXX 010X XXXX 011X
9	XMT Buffer Empty	XXXX 000X
10	Handshake Transition	XXXX 001X
Lowest		

Figure 3-6. Interrupt Vectors and Priority (7806/7)

If properly selected, the vectors for the CTC and DART will be consecutive (2 bytes apart). If the CTC vectors are to precede the DART vectors, program the CTC with a vector of X8 hex and the DART with a vector of (X+1)0 hex. The vectors will run from hex X8 to (X+1)E. If the DART vectors come first, program the DART with a vector of X0 hex and the CTC with a vector of (X+1)0 hex. The vectors will run from hex X0 to (X+1)6.

There are three circumstances in which the interrupts should be handled differently from that previously explained.

1. Interrupts are not used at all, for whatever reason.

   Monitor the Interrupt Pending bit of Status 0. When set, read the vector from Status 2 (having previously initialized the Status Affects Vector mode). Use the vector to determine the appropriate response and use the Return From Interrupt command of Control 0 after servicing the DART. The Return From Interrupt command re-enables the DART priority chain.

2. Other cards in the system generate interrupts but only provide 8080-type restart vectors.

   Use Z80 interrupt mode 0 and program the DART with a restart instruction for a vector. Do not use the Status Affects Vector mode. Use the Z80 Return From Interrupt command after servicing the DART.

3. Other cards in the system generate interrupts but do not provide vectors.

   Use Z80 interrupt mode 1. Upon interrupt, check the various possible sources of the interrupt. This can be done on the DART by checking the Interrupt Pending bit of Status 0. If the DART has generated the interrupt, check the XMT Buffer Empty and RCV Character Available flags. If neither of these is set, check the Handshake and Break flags. All of these are in Status 0. If the RCV Character Available flag is set, check the Error flags of Status 1 to see if there were any errors in the reception. Use the Z80 Return From Interrupt command or the Return From Interrupt command of Control 0 after servicing the DART. This re-enables the priority chain within the DART.

   The Status Affects Vector mode and the vector readable in Status 2 cannot be used in this mode. The modified vector is only available while the interrupt is pending and would already have returned to its initial condition before the program could check it.

Handshaking

Three elements control handshaking for the serial and parallel ports: the DART, the processor, and jumpers (serial only). The jumpers select which RS-232-C line each handshake pin of the DART is connected to (see figure 2-4, page 2-5). The DART can control some of the handshake lines, and the processor can control all or some of them through the DART registers, dependent on whether auto handshaking is enabled.

Figure 3-7 shows which lines can be controlled by the DART.

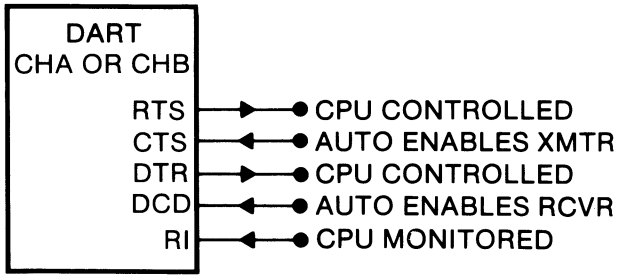


Figure 3-7. DART Handshaking Capability (7806/7)

The CTS* and DCD* pins automatically enable/disable the transmitter and receiver if auto handshaking is selected. If the receiver or transmitter is disabled while receiving/transmitting a character, the character may be lost. The character should be repeated. Perform a Channel Reset command through Control 0, reinitialize, and repeat the character.

If auto handshaking is not enabled, the CTS* and DCD* pins must be monitored by the processor or they can generate interrupts if handshake interrupts are enabled. The RI* line must also be monitored by the processor or can generate interrupts. RI*, CTS*, and DCD* are monitored through Status 0.

The DTR* and RTS* pins are controlled by the processor through Control 5.

Block Transfers

Block transfers can be between the DART and the CPU or the DART and a DMA controller. Block transfers are controlled by the two Wait/Ready lines (W/Rdy A* and W/Rdy B*), one for each channel. They are accessible at the connector for the CTC (figures 2-2 and 2-23, pages 2-2 and 2-22).

For DMA block transfers, the Wait/Ready line should be programmed in the Ready mode (see the description of bits 5, 6, and 7 of Control 1). In this mode, the line indicates to the DMA controller when the DART is ready to transfer another byte.

For CPU block transfers, the Wait/Ready line should be programmed in the Wait mode. The line should be connected to the STD BUS WAITRQ* line. The Wait/Ready line can be enabled at initialization and left enabled if all transfers will be block transfers. Since the line only becomes active when the processor attempts to read or write a character before the DART is ready, it does not interfere with normal processor operation. Reading from or writing to the Control or Status registers is not affected by the Wait/Ready line.

The CPU block transfers are performed using the Z80 block I/O transfer instructions. Blocks of up to 256 bytes can be transferred. The Interrupt on First Character mode should be used when receiving in the Block mode. In the Wait mode, the Wait/Ready line is open drain, so use a pullup resistor on it.

7807 PARALLEL PORT

The handshake lines for the 7807 parallel port are controlled by the DART as described in Serial Channel and Parallel Handshake, page 3-2. The eight bits of the parallel output port are latched by writing data to the parallel port (address FD hex as shipped). Both the handshake and data lines are accessed at connector J2. Figure 2-22, page 2-21 shows the pinout of connector J1.

The parallel port can be used in two ways. The port can be used to provide a communications interface for Centronics-type printers. Typical software to accomplish this is provided in section 4. Information on cabling for communications with a Centronics-type printer is given in figure 2-22, page 2-21. The parallel port can also be used on an individual-bit basis to provide general-purpose control. The handshake lines can also be used to provide this control, resulting in nine output bits and two input bits. Two-pin connector cables, such as those indicated in figure 2-3, page 2-4, can be used in this bit control. This bit control would be used in applications such as lighting LEDs, closing relays, and detecting contact closure.

COUNTER/TIMER

The Counter/Timer Chip (CTC) can be used for a number of purposes including: timing (based on the onboard clock), counting (of external events), inputting external interrupts (to prioritize them and generate unique vectors), generating the baud rate clock(s) for the DART, and generating timed output pulses.

All four channels can generate an interrupt when they reach a zero count. Three of the channels also have an output line which indicates when a zero count has been reached. One of these channels has its output line dedicated to baud rate generation. On the 7806, a second CTC channel may also be used for baud rate generation. As the CTC is implemented on the 7806/7, functions for each channel are shown in figure 3-8.

	CHANNELS WHICH CAN BE USED FOR EACH FUNCTION		
FUNCTION	NO BAUD CLOCKS	1 BAUD CLOCK	2 BAUD CLOCKS
External Inputs (for counting or Interrupt inputs)	1,2,3	1,2,3	2,3
Timers	0,1,2,3	1,2,3	2,3
Output Signals	1,2	1,2	2

Figure 3-8. CTC Channels Per Function (7806/7)

The internal structure of the CTC is shown in figure 3-9. Each channel has a Control register, an Initial Count or Load register, and a Current Count register used to read the counter. Channel 0 also has a Vector register. The Control and Current Count registers are directly accessible. The Load registers and the Channel 0 Vector register are addressed through the Control registers as explained in the following pages.

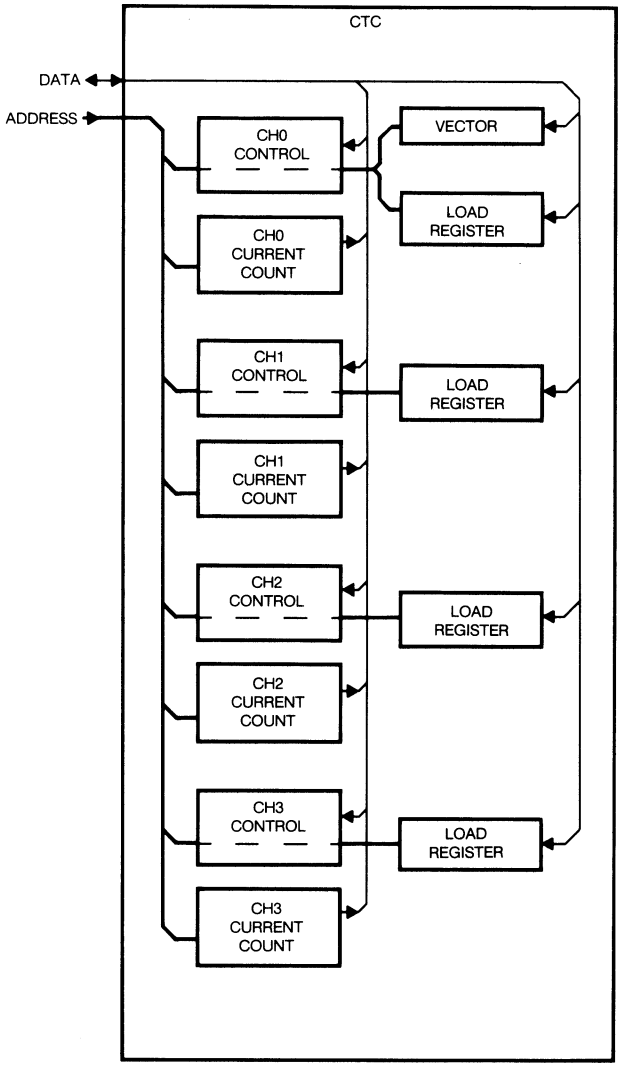


Figure 3-9. CTC Internal Structure (7806/7)

Control Registers

The Control register is the same for each channel and is shown in figure 3-10.

BIT							
7	6	5*	4	3*	2	1	0
ENABLE INTS	COUNT-ER/TIMER	CLOCK DIVIDER	CLOCK EDGE	TRIGGER	LOAD REG.	CHANNEL RESET	LOAD VECTOR

BIT	DESCRIPTION
7	1=Enable Interrupts
6	1=Counter Mode 0=Timer Mode
5*	1=Clock divided by 256 0=Clock divided by 16
4	1=External clock/trigger line is rising edge sensitive 0=Falling edge sensitive
3*	1=Timer started by external trigger 0=Timer starts after being loaded
2	1=Next byte to this channel is load register data
1	1=Channel stops; must be reloaded or re-initialized Momentary function
0	Channel 0 only: 0=byte to be loaded to vector register

*Timer mode only

Figure 3-10. CTC Control Registers (7806/7)

Load Vector (B0): Bit 0 must always be set to a one when writing to a control register, except when writing the vector. If the bit is set to a zero when writing to the Channel 0 Control register, the byte is loaded directly into the Vector register. It has no effect on the Control register. A byte written to any other Control register with bit 0 equal to 0 will have an undefined effect.

Channel Reset (B1): This bit stops the counter. It does not need to be written back to a zero to continue operation. The channel resumes operation as soon as a new Control and/or Load register byte are written. If both this bit and the Load register bit are set, the channel resumes operation as soon as a new Load register byte is written. This bit does not clear the Control register and, when this bit is written to, the other bits are affected. Therefore, be sure to write the other bits to the desired state.

Load Register (B2): This bit sets up the CTC channel to accept the next byte written to that channel as Load register data.

Trigger (B3): This bit is used only in the Timer mode. It enables the timer to be started by a pulse on the external input for that channel. Otherwise, the timer starts as soon as the count value is loaded.

Clock Edge (B4): With this bit set, the external input is positive edge sensitive. Otherwise, it is negative edge sensitive.

Clock Divider (B5): This bit is used in the Timer mode only. If set, the main CTC clock input is divided by 256 and then drives the timer. If the bit is reset, the clock is divided by 16.

Counter/Timer (B6): If set, the Counter mode is selected. If reset, the Timer mode is selected.

Enble Ints (B7): If set, allows an interrupt to be generated whenever the down counter reaches zero.

Load Register

The Load register is accessed through the Control register. A value must be entered into the Load register before the channel will operate. In the Timer mode, if the timer is not started by an external trigger, it starts immediately after a value is written to the Load register.

The counter is reloaded from the Load register as soon as it reaches a count of zero. If a new value is loaded while the channel is operating, the channel finishes its current count before being loaded with the new count. The lowest value is 01, the highest is 00, which yields 256 counts.

Current Count Register

The Current Count register makes available the value in the counter. It is latching so the value read is always valid. It is not possible to catch the counter in transition and get an invalid reading.

Since the counter is a down counter only, an inversion is required when counting external events or measuring time intervals. Load the channel with FF hex initially. Then invert the number read from the Current Count register to get the actual count.

Vector Register

The Vector register only exists in Channel 0. It is accessed by writing to the Control register with bit 0 cleared. The byte thus written is loaded directly into the Vector register. Bit 0 must always be zero (the Z80 requires this anyway) and bits 1 and 2 are modified by the CTC to provide a unique vector for each channel. Only bits 3 through 7, therefore, are selected by the user.

If properly selected, the vectors for the CTC and DART will be consecutive. If the CTC vectors are to precede the DART vectors, program the CTC with a vector of X8 hex and the DART with a vector of (X+1)0 hex. The vectors will run from hex X8 to (X+1)E. If the DART vectors come first, program the DART with a vector of X0 hex and the CTC with a vector of (X+1)0 hex. The vectors will run from hex X0 to (X+1)6.

CTC Operation

To operate the CTC, simply write to the Control register and Load register and it is ready to operate. If interrupts are used, the Vector register must also be initialized. The CTC interrupts can only be used in the Z80 interrupt mode 2.

Interrupts: The internal interrupt priority of the CTC and the vectors generated by each channel are shown in figure 3-6, page 3-13.

The CTC can be used as an external interrupt prioritizer. Use the CTC in the Counter mode and place a count of one in it. Every time an active edge is received on the external input line, the count goes to zero and an interrupt is generated.

Care must be taken when disabling the CTC interrupts. It is possible to disable the CTC interrupts just as an interrupt is being generated. In this case, the CTC may not provide the Interrupt vector when it is requested by the processor. For this reason, the Z80 interrupts should be temporarily disabled just before disabling the CTC interrupts, then re-enabled.

The Z80 Return From Interrupt command should always be used at the end of an interrupt service routine. This re-enables the interrupt priority chain.

Cascading: Channels can be cascaded together to give higher counts or longer time delays. To cascade, place one channel in the Counter mode and connect its external input to the output signal of another channel. This other channel may be in the Counter or Timer mode. Each time it reaches a count of zero, it decrements the channel whose input it is driving.

Output Pulses: The CTC can be used to generate a series of pulses since the output line goes low momentarily each time the counter reaches zero. The frequency is a function of the count value and the clock divider.

Software Count/Trigger: Once a channel has been initialized, it can be decremented or triggered by software. In the Counter mode, the processor can decrement the counter; in the Timer mode, the processor can act as a trigger to start the timer.

To do this, write a new Control Register byte with the Clock Edge bit complemented. Once a channel has been initialized, any change in this bit, besides changing the polarity of the external input, also acts as a pulse on the input. This can be used to decrement a counter or start a timer.

CTC Timing: Detailed timing for the CTC can be found in data sheets on Zilog's Z8430A and Mostek's MK3882-4. Listed here are the main points on CTC timing which should be considered before operation.

- An additional cycle is not required for the zero count. As an example, if loaded with 4, the down count would be 4, 3, 2, 1, 4, 3, ...
- The output line goes low at zero count, but does not stay low. It returns high after approximately one clock cycle, even in the counter mode.
- The incoming count pulses must not be more than half the CTC clock rate (4 MHz for 7806/7-0, 3.6864 MHz for 7806/7-1, 2.5 MHz for 7806/7-2).

- When a count pulse is received, the down counter decrements on the next rising edge of the CTC clock. If the count pulse occurs too close to the rising edge of the clock, the decrement occurs on the next rising edge of the clock.

- The timer starts, or is ready for a trigger pulse, after the machine cycle of the instruction which writes to the CTC. What is written may be a new Control word or a Load register word. The new Control word does not specify a Load register word to follow. Specifically, the timer is enabled on the rising edge of the second clock pulse after the machine cycle.

- If the timer is started by external trigger, it starts on the second rising clock edge after the trigger. If the trigger occurred too close to a rising edge, the timer starts after the third rising clock edge.

- If a Control word is to be followed by a Load register word, the timer trigger is not enabled until the Load register word is written.

- External inputs must remain high or low at least 200 nanoseconds.

INTRODUCTION

This section contains hardware-level subroutine modules with which to operate the 7806 and 7807 cards. The software in this section can be used without license from Pro-Log. Although tested and believed correct, this software is not represented to be free from errors or copyright infringement, or appropriate for any specific application.

The subroutines are in STD instruction mnemonics. The coding forms are grouped at the end of this section, following the flowcharts and subroutine specifications.

MEMORY ADDRESSES

Full memory addresses are given. They are preferred addresses that allow the subroutines to work with those provided for other Series 7000 STD BUS cards from Pro-Log. The program addresses correspond to the 7806/7 onboard memory sockets.

If your system cannot use the memory addresses in the 7806 or 7807 software packages, simply change the memory page addresses as required when loading these modules into your system.

I/O PORT ADDRESSES

The 7806/7 I/O ports are assigned preferred hexadecimal addresses F0-F7 and FE, the 7807 also uses FD for compatibility with other Series 7000 cards. Section 2 shows how to remap these addresses if necessary. This software can be used by simply changing the port addresses when loading the program modules into your system.

DESCRIPTION

The software in this section can be used to operate both the DART and CTC. It is written mainly for the DART, but the subroutine which initializes the CTC for baud rate generation may be used for general-purpose CTC control.

There are 17 7806 subroutines and 18 7807 subroutines, most of which are subroutines within subroutines and are not accessed directly. Only five 7806 and three 7807 subroutines need to be accessed directly to perform serial communication; two 7807 subroutines may be accessed directly for a print operation through a Centronics-type printer. Another six 7806 and seven 7807 subroutines may be accessed and could be useful as building blocks in the user's own software for data communication or CTC control. All of these are shown in figure 4-1 for the 7806 and figure 4-23, page 4-26, for the 7807.

There is a separate software section for the 7806 and 7807. Many of the subroutines are very similar or identical. The subroutines have been separated into two sections to avoid confusion.

This software is not complete in itself. There is no software provided to process errors in data reception or changes in incoming serial handshake lines since these are largely application dependent. If used as a package with user-supplied software, these subroutines can greatly simplify the use of the serial and parallel ports.

7806 SOFTWARE

NAME	DESCRIPTION	MAIN SUB-ROUTINE	MAY BE USED BY ITSELF	NOT USEFUL BY ITSELF
(INIT COMM)	Initialize Communications Interface	X		
(INIT INTER)	Initialize Z80 Interrupt		X	
(INIT CTC 0)	Initialize CTC Channel 0		X	
(INIT CTC 1)	Initialize CTC Channel 1		X	
(INIT DART)	Initialize entire DART		X	
(INIT DART CH)	Initialize one channel of DART		X	
(XMT BLK B)	Transmit a block of data (CH B)			X
(XMT BLK A)	See above (CH A)			X
(XMT CHAR)	Transmit a character			X
(STRT XMT A)	Start block transmission (CH A)	X		
(STRT XMT B)	See above (CH B)	X		
(XMT OFF)	Turn off transmitter interrupt at end of block		X	
(END XMT B?)	Verify that entire block has been transmitted (CH B)			X
(END XMT A?)	See above (CH A)			X
(RCV & ECHO A)	Receive and echo a character (CH A)	X		
(RCV & ECHO B)	See above (CH B)	X		
(RCV & ECHO)	Receive and echo a character			X

Figure 4-1. 7806 Subroutine Types

If used as a package (accessing just the five main 7806 subroutines), only a few requirements must be met by the user. The five main subroutines and their requirements are given in figure 4-2. The following list details the requirements, and the points to be considered before implementing the software.

- The user may want to provide software for handshake control or for responding to changes in incoming handshaking.
- The software does not provide for response to errors in received data.
- The software does not store or respond to received data. The character is simply received and echoed. A Jump to Subroutine instruction is provided to jump to the user's own subroutine (PROCESS CHAR). The user's subroutine may then store and/or respond to the character according to the requirements of the application.
- The jump to (PROCESS CHAR) may occur before or after the character is echoed (for after the echo, addresses hex 1E58 for Channel A, 1E48 for Channel B, and for before the echo, address hex 1E57).
- This software operates with Z80 interrupt mode 2. It selects mode 2 and enables the interrupt as part of the initialization procedure. It also loads the Z80 interrupt register. The data to be loaded into the register must be selected by the user (see address hex 1D1B).
- The baud rates and various other factors selected by the initialization routine should be checked to see that they match the application. The initialized values are given in the description of the (INIT COMM) routine.
- The number of bytes to be transferred must be in register A (FF = 255) and the address of the data to be transferred in the HL register pair before using the transmit routines.
- Keep in mind the 7806 onboard interrupt priority (figure 3-6, page 3-13).

NAME	DESCRIPTION	USE REQUIREMENTS
(INIT COMM)	Initialize Communications Interface (Entry Address 1D00 hex)	None
(STRT XMT A)	Transmit a block of data (CH A) (Entry Address 1DC0 hex)	Data must be in memory, data pointer in register pair HL, and number of bytes in register A
(STRT XMT B)	See above (CH B). (Entry Address 1DE0 hex)	See above
(RCV & ECHO A)	Receive and echo a character (CH A) (Entry Address 1E40 hex)	User must provide software to store and/or respond to the received character.
(RCV & ECHO B)	See above (CH B). (Entry Address 1E50 hex)	See above

Figure 4-2. 7806 Main Subroutines Use Requirements

7806 SUBROUTINE SPECIFICATIONS: (INIT COMM)

ENTRY ADDRESS	:	1D00
MEMORY REQUIREMENTS		
Program Memory	:	21 (93 with subroutines and tables)
RAM	:	0
Stack	:	2
REGISTERS ALTERED	:	A, F, B, C, H, L
SUBROUTINES USED	:	(INIT CTC 0), (INIT CTC 1), (INIT DART), (INIT INTER)
EXECUTION TIME	:	1064 Time States
ENTRY REQUIREMENTS	:	None
EXIT CONDITIONS	:	N/A
FUNCTIONAL DESCRIPTION	:	

This subroutine initializes the DART, the CTC (to generate baud rates), and the Z80 interrupt. The initialized conditions should be checked to be sure they match the application. The initialized values are shown in figure 4-3. Refer to figures 3-4 and 3-10, pages 3-4 and 3-18, while reviewing these. The initial baud rates are 9600 for Channel A and 300 for Channel B using the 4 MHz version (7806-0).

ADDRESS	FUNCTION	DATA
1D21	CTC CH 0 Control Register	0100 0101
1D25	CTC CH 0 Load Register	0000 1101
1D31	CTC CH 1 Control Register	0000 0101
1D35	CTC CH 1 Load Register	0011 0100
1D63	DART CH B Control Register 2	0000 0000
1D65	DART CH B Control Register 4	0100 1111
1D67	DART CH B Control Register 5	1010 1010
1D69	DART CH B Control Register 3	0110 0001
1D6B	DART CH B Control Register 1	0001 0100
1D73	DART CH A Control Register 4	0100 1111
1D75	DART CH A Control Register 5	1010 1010
1D77	DART CH A Control Register 3	0110 0001
1D79	DART CH A Control Register 1	0001 0100

Figure 4-3. Initialization Values (7806)

7806 SUBROUTINE SPECIFICATIONS: (INIT INTER)

ENTRY ADDRESS	:	1D18
MEMORY REQUIREMENTS		
Program Memory	:	8
RAM	:	0
Stack	:	0
REGISTERS ALTERED	:	A, I
SUBROUTINES USED	:	None
EXECUTION TIME	:	38 Time States
ENTRY REQUIREMENTS	:	None
EXIT CONDITIONS	:	N/A
FUNCTIONAL DESCRIPTION	:	

This subroutine selects Z80 interrupt mode 2 and enables the Z80 interrupt. It also loads the Z80 interrupt register. The byte to be loaded in the interrupt registers must be provided by the user and stored at address hex 1D1B.

7806 SUBROUTINE SPECIFICATIONS: (INIT CTC 0)

ENTRY ADDRESS	:	1D20
MEMORY REQUIREMENTS		
Program Memory	:	9
RAM	:	0
Stack	:	0
REGISTERS ALTERED	:	A
SUBROUTINES USED	:	None
EXECUTION TIME	:	46 Time States
ENTRY REQUIREMENTS	:	None
EXIT CONDITIONS	:	N/A
FUNCTIONAL DESCRIPTION	:	

This subroutine initializes the CTC Channel 0. The Vector register is not loaded. Refer to figure 4-3 for the initialized values.

7806 SUBROUTINE SPECIFICATIONS: (INIT CTC 1)

ENTRY ADDRESS	:	1D30
MEMORY REQUIREMENTS		
Program Memory	:	9
RAM	:	0
Stack	:	0
REGISTERS ALTERED	:	A
SUBROUTINES USED	:	None
EXECUTION TIME	:	46 Time States
ENTRY REQUIREMENTS	:	None
EXIT CONDITIONS	:	N/A
FUNCTIONAL DESCRIPTION	:	

This module initializes CTC Channel 1. Refer to figure 4-3 for the initialized values.

7806 SUBROUTINE SPECIFICATIONS: (INIT DART)

ENTRY ADDRESS	:	1D40
MEMORY REQUIREMENTS		
Program Memory	:	13 (47 with subroutine and table)
RAM	:	0
Stack	:	2
REGISTERS ALTERED	:	A, F, B, C, H, L
SUBROUTINES USED	:	(INIT DART CH)
EXECUTION TIME	:	826 Time States
ENTRY REQUIREMENTS	:	None
EXIT CONDITIONS	:	N/A
FUNCTIONAL DESCRIPTION	:	

This subroutine initializes both channels of the DART. Refer to figure 4-3 for the initialized values.

7806 SUBROUTINE SPECIFICATIONS: (INIT DART CH)

ENTRY ADDRESS	:	1D50
MEMORY REQUIREMENTS		
Program Memory	:	12 (plus 11 or 22 byte table for one or two channels)
RAM	:	0
Stack	:	0
REGISTERS ALTERED	:	A, B, C, H, L, F
SUBROUTINES USED	:	None
EXECUTION TIME	:	420 Time States maximum
ENTRY REQUIREMENTS		
Register Pair HL	:	Points to CH. A data table (1D70) or CH. B data table (1D60)
EXIT CONDITIONS	:	N/A
FUNCTIONAL DESCRIPTION	:	

This subroutine initializes one channel of the DART. The channel is selected by loading register pair HL with the appropriate address (see Entry Requirements). Refer to figure 4-3 for the initialized values.

7806 SUBROUTINE SPECIFICATIONS: (STRT XMT A)

ENTRY ADDRESS	:	1DC0
MEMORY REQUIREMENTS		
Program Memory	:	26 (93 with subroutines)
RAM	:	3
Stack	:	16
REGISTERS ALTERED	:	None
SUBROUTINES USED	:	(END XMT A?), (XMT BLK A), (XMT CHAR), (XMT OFF)
EXECUTION TIME:	:	443 Time States (if transmission of previous block has been completed)
ENTRY REQUIREMENTS		
Register A	:	Number of bytes to be sent (FF = 255)
Register Pair HL	:	Address of first byte to be sent
Memory	:	Data to be sent
EXIT CONDITIONS	:	Byte count decremented and stored at hex 3200, data pointer incremented and stored at 3201 and 3202
FUNCTIONAL DESCRIPTION	:	

This subroutine transmits a block of data of up to 255 bytes. The program first checks to see if a previous block is still in the process of being sent. If it is, the program stays in a continuous checking loop. When the block can be sent, the first byte is sent and the program exits. Thereafter, whenever the transmitter indicates it is ready, via interrupt, the program responds by sending another byte. When the transmitter interrupts after sending the last byte, the program turns off the transmitter interrupt.

7806 SUBROUTINE SPECIFICATIONS: (STRT XMT B)

ENTRY ADDRESS	:	1DE0
MEMORY REQUIREMENTS		
Program Memory	:	26 (93 with subroutines)
RAM	:	3
Stack	:	16
REGISTERS ALTERED	:	None
SUBROUTINES USED	:	(END XMT B?), (XMT BLK B), (XMT CHAR), (XMT OFF)
EXECUTION TIME	:	443 Time States (if transmission of previous block has been completed)
ENTRY REQUIREMENTS		
Register A	:	Number of bytes to be sent (FF = 255)
Register Pair HL	:	Address of first byte to be sent
Memory	:	Data to be sent
EXIT CONDITIONS	:	Byte count decremented and stored at hex 3203, data pointer incremented and stored at 3204 and 3205
FUNCTIONAL DESCRIPTION	:	

This subroutine transmits a block of data of up to 255 bytes. The program first checks to see if a previous block is still in the process of being sent. If it is, the program stays in a continuous checking loop. When the block can be sent, the first byte is sent and the program exits. Thereafter, whenever the transmitter indicates it is ready, via interrupt, the program responds by sending another byte. When the transmitter interrupts after sending the last byte, the program turns off the transmitter's interrupt.

7806 SUBROUTINE SPECIFICATIONS: (RCV & ECHO A)

ENTRY ADDRESS	:	1E40
MEMORY REQUIREMENTS		
Program Memory	:	15 (21 with subroutines)
RAM	:	0
Stack	:	6
REGISTERS ALTERED	:	None
SUBROUTINES USED	:	(RCV & ECHO), (PROCESS CHAR) "PROCESS CHAR" must be supplied by the user
EXECUTION TIME	:	149 Time States
ENTRY REQUIREMENTS	:	None
EXIT CONDITIONS	:	N/A
FUNCTIONAL DESCRIPTION	:	

This is an interrupt service routine. It receives a character and immediately echoes it back on the same channel. The user must supply a program (PROCESS CHAR) to interpret and/or store the received character. A Jump to Subroutine instruction is provided at address 1E48 for this purpose. This will jump to the (PROCESS CHAR) subroutine after the character has been echoed. Alternatively, a Jump to Subroutine instruction may be placed at address 1E62 which is before the character is echoed. Note that this point in the program is common to (RCV & ECHO A) and (RCV & ECHO B). The receivers are enabled at initialization and, since reception is interrupt driven, the (PROCESS CHAR) routine must flag the processor that a message has been received. Otherwise, data reception is completely transparent to the processor.

7806 SUBROUTINE SPECIFICATIONS: (RCV & ECHO B)

ENTRY ADDRESS	:	1E50
MEMORY REQUIREMENTS		
Program Memory	:	15 (21 with subroutines)
RAM	:	0
Stack	:	6
REGISTERS ALTERED	:	None
SUBROUTINES USED	:	(RCV & ECHO), (PROCESS CHAR) "PROCESS CHAR" must be supplied by the user
EXECUTION TIME	:	149 Time States
ENTRY REQUIREMENTS	:	None
EXIT CONDITIONS	:	N/A
FUNCTIONAL DESCRIPTION	:	

This is an interrupt service routine. It receives a character and immediately echoes it back on the same channel. The user must supply a program (PROCESS CHAR) to interpret and/or store the received character. A Jump to Subroutine instruction is provided at address 1E58 for this purpose. This will jump to the (PROCESS CHAR) subroutine after the character has been echoed. Alternatively, a Jump to Subroutine instruction may be placed at address 1E62 which is before the character is echoed. Note that this point in the program is common to (RCV & ECHO A) and (RCV & ECHO B). The receivers are enabled at initialization and, since reception is interrupt driven, the (PROCESS CHAR) routine must flag the processor that a message has been received. Otherwise, data reception is completely transparent to the processor.

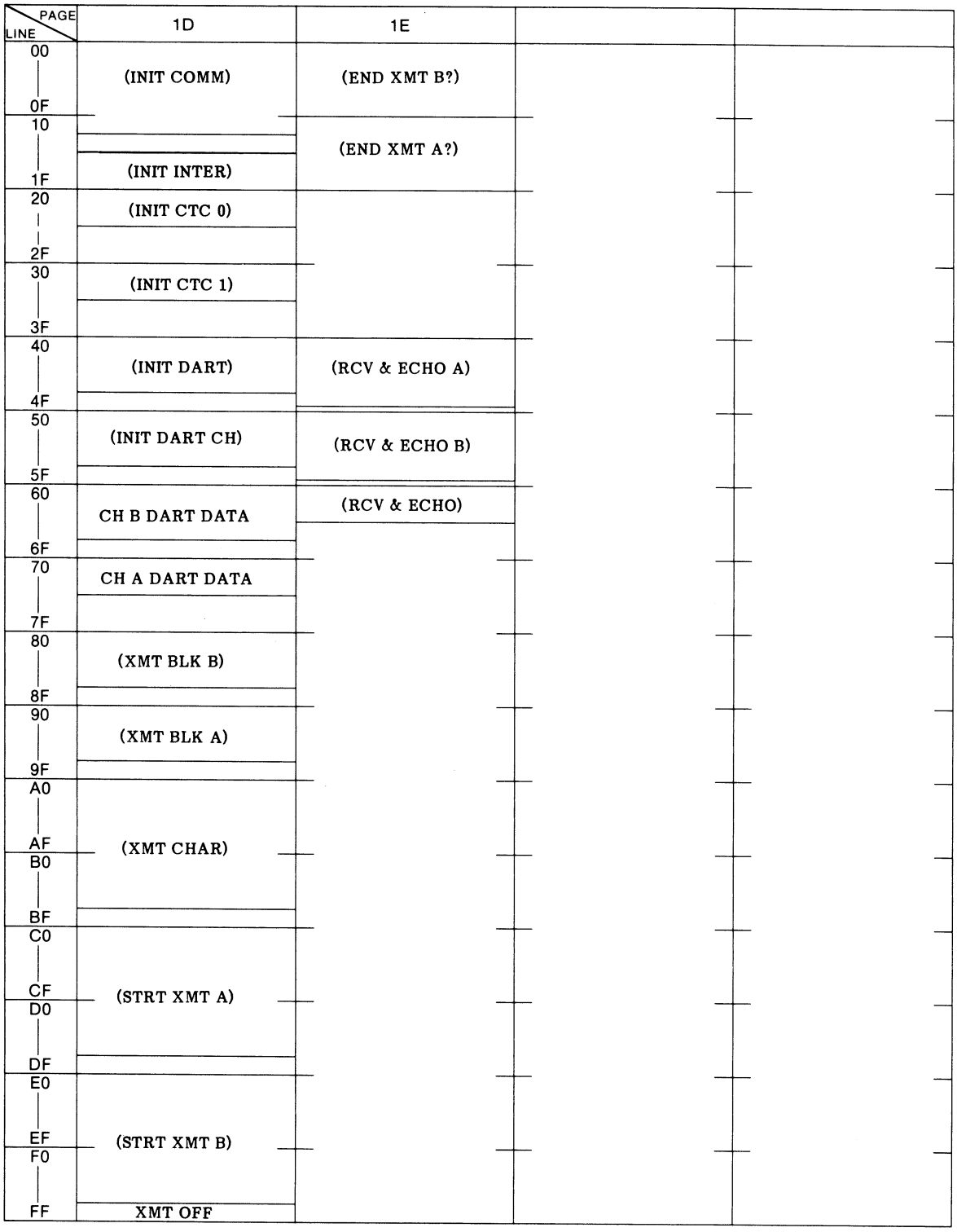


Figure 4-4. 1K Program Memory Map (7806)

PAGE ADDRESS: 32

LINE	LABEL
00	XMT BYTE COUNT A
01	DATA LSB
02	POINTER A MSB
03	XMT BYTE COUNT B
04	DATA LSB
05	POINTER B MSB
06	
07	
08	
09	
0A	
0B	
0C	
0D	
0E	
0F	
10	
11	
12	
13	
14	
15	
16	
17	
18	
19	
1A	
1B	
1C	
1D	
1E	
1F	
20	
21	
22	
23	
24	
25	
26	
27	
28	
29	
2A	
2B	
2C	
2D	
2E	
2F	
30	
31	
32	
33	
34	
35	
36	
37	
38	
39	
3A	
3B	
3C	
3D	
3E	
3F	

LINE	LABEL
40	
41	
42	
43	
44	
45	
46	
47	
48	
49	
4A	
4B	
4C	
4D	
4E	
4F	
50	
51	
52	
53	
54	
55	
56	
57	
58	
59	
5A	
5B	
5C	
5D	
5E	
5F	
60	
61	
62	
63	
64	
65	
66	
67	
68	
69	
6A	
6B	
6C	
6D	
6E	
6F	
70	
71	
72	
73	
74	
75	
76	
77	
78	
79	
7A	
7B	
7C	
7D	
7E	
7F	

LINE	LABEL
80	
81	
82	
83	
84	
85	
86	
87	
88	
89	
8A	
8B	
8C	
8D	
8E	
8F	
90	
91	
92	
93	
94	
95	
96	
97	
98	
99	
9A	
9B	
9C	
9D	
9E	
9F	
A0	
A1	
A2	
A3	
A4	
A5	
A6	
A7	
A8	
A9	
AA	
AB	
AC	
AD	
AE	
AF	
B0	
B1	
B2	
B3	
B4	
B5	
B6	
B7	
B8	
B9	
BA	
BB	
BC	
BD	
BE	
BF	

LINE	LABEL
C0	
C1	
C2	
C3	
C4	
C5	
C6	
C7	
C8	
C9	
CA	
CB	
CC	
CD	
CE	
CF	
D0	
D1	
D2	
D3	
D4	
D5	
D6	
D7	
D8	
D9	
DA	
DB	
DC	
DD	
DE	
DF	
E0	
E1	
E2	
E3	
E4	
E5	
E6	
E7	
E8	
E9	
EA	
EB	
EC	
ED	
EE	
EF	
F0	
F1	
F2	
F3	
F4	
F5	
F6	
F7	
F8	
F9	
FA	
FB	
FC	
FD	
FE	
FF	

Figure 4-5. 256-Byte RAM Memory Map (7806)

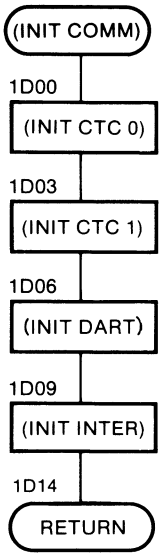


Figure 4-6. (INIT COMM) Flowchart (7806)

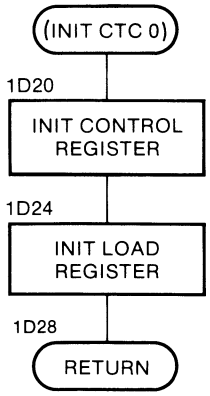


Figure 4-7. (INIT CTC 0) Flowchart (7806)

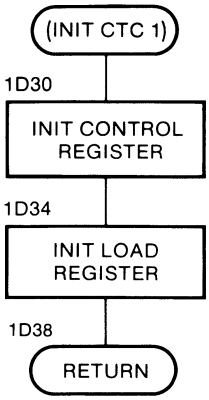


Figure 4-8. (INIT CTC 1) Flowchart (7806)

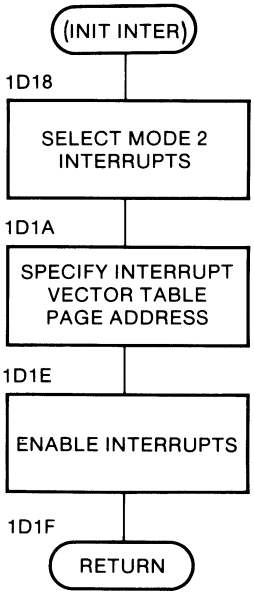


Figure 4-9. (INIT INTER) Flowchart (7806)

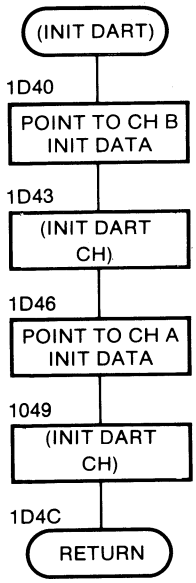


Figure 4-10. (INIT DART) Flowchart (7806)

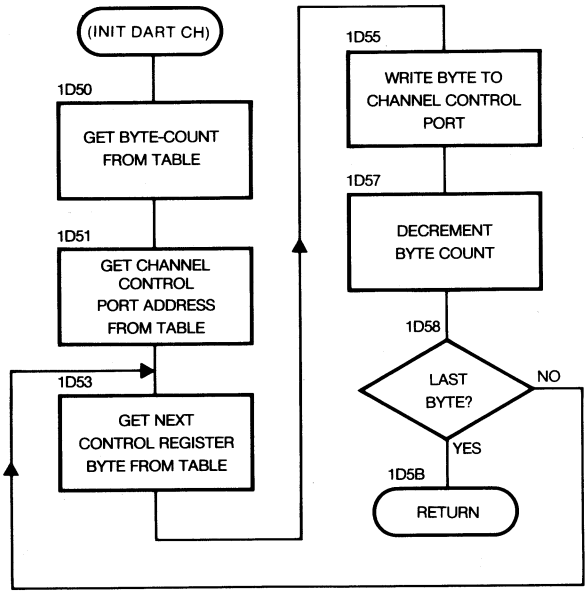


Figure 4-11. (INIT DART CH) Flowchart (7806)

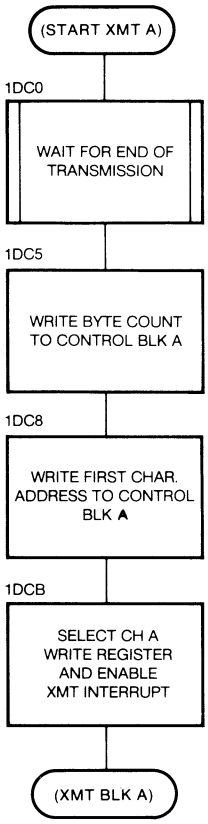


Figure 4-12. (STRT XMT A) Flowchart (7806)

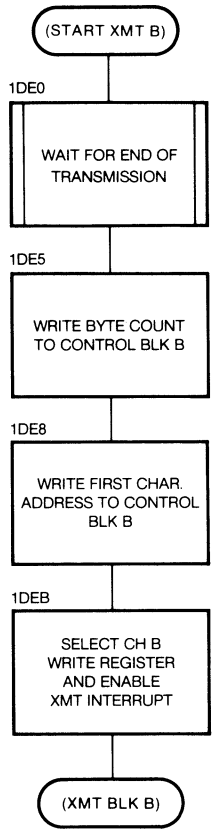


Figure 4-13. (STRT XMT B) Flowchart (7806)

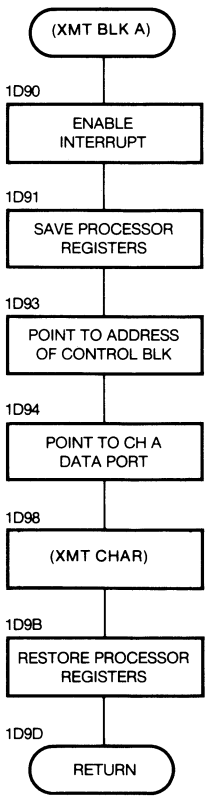


Figure 4-14. (XMT BLK A) Flowchart (7806)

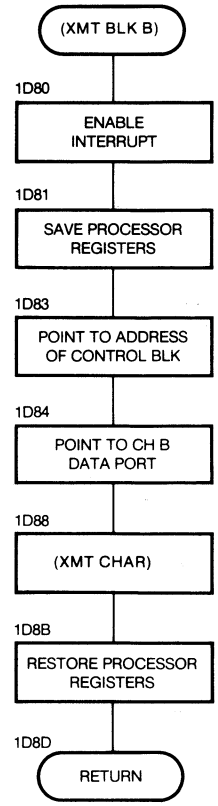


Figure 4-15. (XMT BLK B) Flowchart (7806)

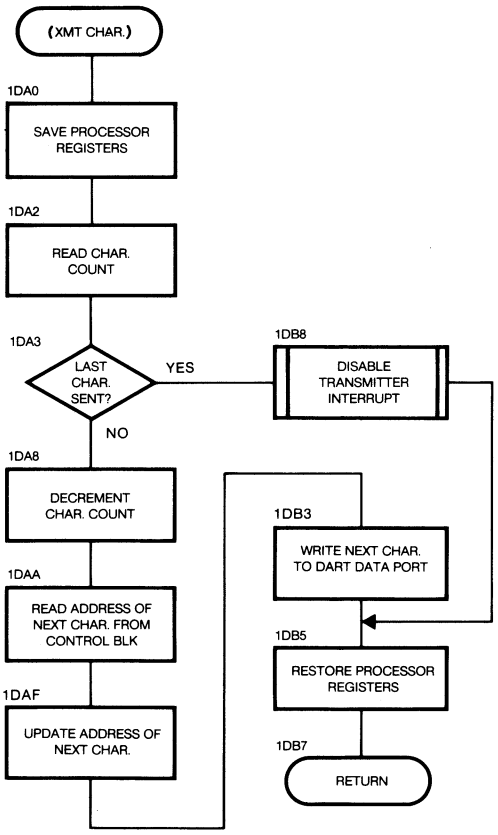


Figure 4-16. (XMT CHAR) Flowchart (7806)

Figure 4-17. (END XMT?) Flowchart (7806)

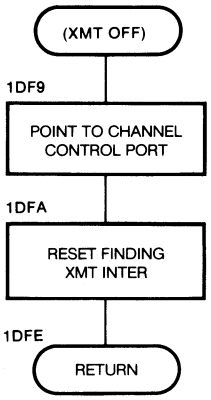


Figure 4-18. (XMT OFF) Flowchart (7806)

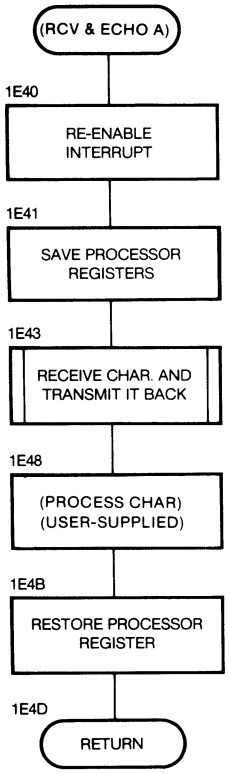


Figure 4-19. (RCV & ECHO A) Flowchart (7806)

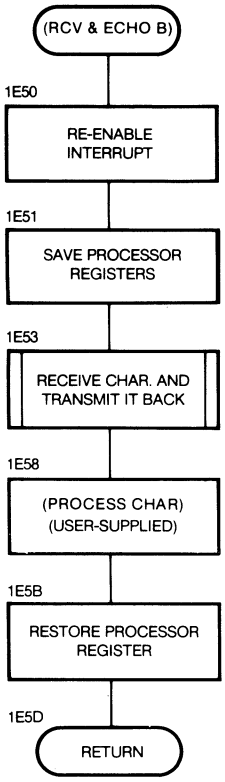


Figure 4-20. (RCV & ECHO B) Flowchart (7806)

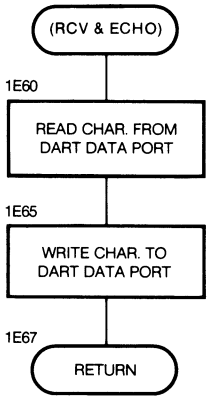


Figure 4-21. (RCV & ECHO) Flowchart (7806)

```
ADRdata!     LABEL      !INSTR !    MODIFIER    !          COMMENTS
-------!---------------!------!----------------!-----------------------------
1D00 CD!(INIT COMM)     !JS    !UNCONDITIONAL   ! INITIALIZE CTC CHAN 0
  01 20!                ! ---- !(INIT CTC 0)    !   BAUD RATE 9600 CH A
  02 1D!                ! ---- !                !
  03 CD!                !JS    !UNCONDITIONAL   ! INITIALIZE CTC CHAN 1
  04 30!                ! ---- !(INIT CTC 1)    !   BAUD RATE 300 CH B
  05 1D!                ! ---- !                !
  06 CD!                !JS    !UNCONDITIONAL   ! INITIALIZE DART CHANNELS A & B
  07 40!                ! ---- !(INIT DART)     !
  08 1D!                ! ---- !                !
  09 CD!                !JS    !UNCONDITIONAL   ! INITIALIZE INTERRUPT STRUCTURE
  0A 18!                ! ---- !(INIT INTER)    !
  0B 1D!                ! ---- !                !
  0C 3E!                !LDAI  !                ! INITIALIZE DART CONTROL BLOCK
  0D 00!                ! ---- !00              !   COUNTS AT 00
  0E 32!                !STAD  !                !
  0F 00!                ! ---- !adr 3200        !
1D10 32!                ! ---- !                !
  11 32!                !STAD  !                !
  12 03!                ! ---- !adr 3203        !
  13 32!                ! ---- !                !
  14 C9!   ----------   !RTS   !UNCONDITIONAL   ! RETURN
  15 FF!                !       !               !
  16 FF!                !       !               !
  17 FF!                !       !               !
1D18 ED!(INIT INTER)    !SEIM 2!                ! SELECT MODE 2 INTERRUPTS
  19 5E!                ! ---- !                !
  1A 3E!                !LDAI  !                ! SPECIFY INTERRUPT VECTOR TABLE
  1B --!                ! ---- !                !   PAGE ADDRESS (USER SUPPLIED)
  1C ED!                !LDR   !I,A             !
  1D 47!                ! ---- !                !
  1E FB!                !ENI   !                ! ENABLE INTERRUPTS
  1F C9!   ----------   !RTS   !UNCONDITIONAL   !
1D20 3E!(INIT CTC 0)    !LDAI  !                ! WRITE CHANNEL CONTROL WORD
  21 45!                ! ---- !CONTROL 0       !
  22 D3!                !OPA   !                !
  23 F0!                ! ---- !CTC CH 0        !
  24 3E!                !LDAI  !                ! WRITE TIME CONSTANT WORD
  25 0D!                ! ---- !0D              !
  26 D3!                !OPA   !                !
  27 F0!                ! ---- !CTC CH 0        !
  28 C9!   ----------   !RTS   !UNCONDITIONAL   ! (INITIALIZED FOR 9600 BAUD)
  29 FF!                !       !               !
  2A FF!                !       !               !
  2B FF!                !       !               !
  2C FF!                !       !               !
  2D FF!                !       !               !
  2E FF!                !       !               !
  2F FF!                !       !               !
1D30 3E!(INIT CTC 1)    !LDAI  !                ! WRITE CHANNEL CONTROL WORD
  31 05!                ! ---- !CONTROL 1       !
  32 D3!                !OPA   !                !
  33 F1!                ! ---- !CTC CH 1        !
  34 3E!                !LDAI  !                ! WRITE TIME CONSTANT WORD
  35 34!                ! ---- !34              !
  36 D3!                !OPA   !                !
  37 F1!                ! ---- !CTC CH 1        !
  38 C9!   ----------   !RTS   !UNCONDITIONAL   ! (INITIALIZED FOR 300 BAUD)
  39 FF!                !       !               !
  3A FF!                !       !               !
  3B FF!                !       !               !
  3C FF!                !       !               !
  3D FF!                !       !               !
  3E FF!                !       !               !
  3F FF!                !       !               !
```

Figure 4-22. Software Listing (7806)

```
ADRdata!     LABEL       !INSTR !    MODIFIER    !          COMMENTS
-------!----------------!------!----------------!-----------------------------
1D40 21!(INIT DART)      !LDPI  !HL              ! INITIALIZE DART CHANNEL B
  41 60!                 ! ---- !CH B DATA       !
  42 1D!                 ! ---- !                !
  43 CD!                 !JS    !UNCONDITIONAL   !
  44 50!                 ! ---- !(INIT DART CH)  !
  45 1D!                 ! ---- !                !
  46 21!                 !LDPI  !HL              ! INITIALIZE DART CHANNEL A
  47 70!                 ! ---- !CH A DATA       !
  48 1D!                 ! ---- !                !
  49 CD!                 !JS    !UNCONDITIONAL   !
  4A 50!                 ! ---- !(INIT DART CH)  !
  4B 1D!                 ! ---- !                !
  4C C9!   ----------    !RTS   !UNCONDITIONAL   !
  4D FF!                 !      !                !
  4E FF!                 !      !                !
  4F FF!                 !      !                !
1D50 46!(INIT DART CH)   !LDRN  !B,(HL)          ! GET NUMBER OF BYTES TO OUTPUT
  51 23!                 !ICP   !HL              ! GET CHAN CONTROL PORT ADDRESS
  52 4E!                 !LDRN  !C,(HL)          !
1D53 23!NXT BYTE         !ICP   !HL              ! OUTPUT TABLE TO CHANNEL CTRL
  54 7E!                 !LDAN  !(HL)            !    FIRST SELECTING WRITE REG
  55 ED!                 !OPAN  !(C)             !    AND THEN OUTPUTTING ITS
  56 79!                 ! ---- !                !    CONTROL BYTE
  57 05!                 !DCR   !B               !
  58 C2!                 !JP    !NOT ZERO        !
  59 53!                 ! ---- !NXT BYTE        !
  5A 1D!                 ! ---- !                !
  5B C9!   ----------    !RTS   !UNCONDITIONAL   !
  5C FF!                 !      !                !
  5D FF!                 !      !                !
  5E FF!                 !      !                !
  5F FF!                 !      !                !
1D60 0A!CH B DATA        !  0A  !                ! NUMBER OF BYTES TO OUTPUT
  61 F7!                 !  F7  !                ! CHANNEL B CONTROL PORT
  62 02!                 !  02  !                ! WRITE REG 2
  63 00!                 !  00  !                ! CONTROL BYTE REG 2
  64 04!                 !  04  !                ! WRITE REG 4
  65 4F!                 !  4F  !                ! CONTROL BYTE REG 4
  66 05!                 !  05  !                ! WRITE REG 5
  67 AA!                 !  AA  !                ! CONTROL BYTE REG 5
  68 03!                 !  03  !                ! WRITE REG 3
  69 61!                 !  61  !                ! CONTROL BYTE REG 3
  6A 01!                 !  01  !                ! WRITE REG 1
1D6B 14!adr 1D6B         !  14  !                ! CONTROL BYTE REG 1
  6C FF!                 !      !                !
  6D FF!                 !      !                !
  6E FF!                 !      !                !
  6F FF!                 !      !                !
1D70 08!CH A DATA        !  08  !                ! NUMBER OF BYTES TO OUTPUT
  71 F5!                 !  F5  !                ! CHANNEL A CONTROL PORT
  72 04!                 !  04  !                ! WRITE REG 4
  73 4F!                 !  4F  !                ! CONTROL BYTE REG 4
  74 05!                 !  05  !                ! WRITE REG 5
  75 AA!                 !  AA  !                ! CONTROL BYTE REG 5
  76 03!                 !  03  !                ! WRITE REG 3
  77 61!                 !  61  !                ! CONTROL BYTE REG 3
  78 01!                 !  01  !                ! WRITE REG 1
1D79 14!adr 1D79         !  14  !                ! CONTROL BYTE REG 1
  7A FF!                 !      !                !
  7B FF!                 !      !                !
  7C FF!                 !      !                !
  7D FF!                 !      !                !
  7E FF!                 !      !                !
  7F FF!                 !      !                !
```

Figure 4-22. 7806 (Continued)

```
ADRdata!     LABEL       !INSTR !    MODIFIER    !          COMMENTS
-------!---------------!------!----------------!---------------------------
1D80 FB!(XMT BLK B)      !ENI   !                ! ENABLE INTERRUPTS
  81 C5!                 !PSP   !BC              ! SAVE PROCESSOR REGISTERS
  82 E5!                 !PSP   !HL              !
  83 21!                 !LDPI  !HL              ! POINT TO ADDRESS OF CONTROL
  84 03!                 ! ---- !3203            !   BLOCK
  85 32!                 ! ---- !                !
  86 0E!                 !LDRI  !C               ! POINT TO CHANNEL B DATA PORT
  87 F6!                 ! ---- !F6              !
  88 CD!                 !JS    !UNCONDITIONAL   ! TRANSMIT CHARACTER
  89 A0!                 ! ---- !(XMT CHAR)      !
  8A 1D!                 ! ---- !                !
  8B E1!                 !PLP   !HL              ! RESTORE PROCESSOR REGISTERS
  8C C1!                 !PLP   !BC              !
  8D ED!                 !RTI   !UNCONDITIONAL   ! RETURN
  8E 4D!   ----------    ! ---- !                !
  8F FF!                 !      !                !
1D90 FB!(XMT BLK A)      !ENI   !                ! ENABLE INTERRUPTS
  91 C5!                 !PSP   !BC              ! SAVE PROCESSOR REGISTERS
  92 E5!                 !PSP   !HL              !
  93 21!                 !LDPI  !HL              ! POINT TO ADDRESS OF CONTROL
  94 00!                 ! ---- !3200            !   BLOCK
  95 32!                 ! ---- !                !
  96 0E!                 !LDRI  !C               ! POINT TO CHANNEL A DATA PORT
  97 F4!                 ! ---- !F4              !
  98 CD!                 !JS    !UNCONDITIONAL   ! TRANSMIT CHARACTER
  99 A0!                 ! ---- !(XMT CHAR)      !
  9A 1D!                 ! ---- !                !
  9B E1!                 !PLP   !HL              ! RESTORE PROCESSOR REGISTERS
  9C C1!                 !PLP   !BC              !
  9D ED!                 !RTI   !UNCONDITIONAL   ! RETURN
  9E 4D!   ----------    ! ---- !                !
  9F FF!                 !      !                !
1DA0 F5!(XMT CHAR)       !PSP   !AF              ! SAVE PROCESSOR REGISTERS
  A1 D5!                 !PSP   !DE              !
  A2 7E!                 !LDAN  !(HL)            ! HAS LAST CHARACTER BEEN SENT?
  A3 FE!                 !CPAI  !                !
  A4 00!                 ! ---- !00              !
  A5 CA!                 !JP    !ZERO            !
  A6 B8!                 ! ---- !LAST            !
  A7 1D!                 ! ---- !                !
  A8 3D!                 !DCA   !                ! NO, DECREMENT COUNT & STORE
  A9 77!                 !STAN  !(HL)            !
  AA 23!                 !ICP   !HL              ! READ ADDRESS OF NEXT CHARACTER
  AB 5E!                 !LDRN  !E,(HL)          !   FROM CONTROL BLOCK
  AC 23!                 !ICP   !HL              !
  AD 56!                 !LDRN  !D,(HL)          !
  AE 1A!                 !LDAN  !(DE)            ! READ NEXT CHARACTER
  AF 13!                 !ICP   !DE              ! UPDATE ADDRESS OF NEXT CHAR-
1DB0 72!                 !STRN  !(HL),D          !   ACTER AND STORE IN CONTROL
  B1 2B!                 !DCP   !HL              !   BLOCK
  B2 73!                 !STRN  !(HL),E          !
  B3 ED!                 !OPAN  !(C)             ! WRITE TO DART DATA PORT
  B4 79!                 ! ---- !                !
1DB5 D1!RESTORE          !PLP   !DE              ! RESTORE PROCESSOR REGISTERS
  B6 F1!                 !PLP   !AF              !
  B7 C9!   ----------    !RTS   !UNCONDITIONAL   !
1DB8 CD!LAST             !JS    !UNCONDITIONAL   ! LAST CHARACTER SENT, TURN-OFF
  B9 F9!                 ! ---- !(XMT-OFF)       !   XMT INTERRUPT
  BA 1D!                 ! ---- !                !
  BB C3!                 !JP    !UNCONDITIONAL   ! RESTORE REGISTERS
  BC B5!                 ! ---- !RESTORE         !
  BD 1D!   ----------    ! ---- !                !
  BE FF!                 !      !                !
  BF FF!                 !      !                !
```

Figure 4-22. 7806 (Continued)

```
ADRdata!     LABEL       !INSTR !    MODIFIER     !          COMMENTS
-------!---------------!------!-----------------!----------------------------
1DC0 F5!(START XMT A)    !PSP   !AF               ! WAIT FOR END OF TRANSMISSION
  C1 CD!                 !JS    !UNCONDITIONAL    !   IN PROGRESS
  C2 10!                 ! ---- !(END XMT A?)     !
  C3 1E!                 ! ---- !                 !
  C4 F1!                 !PLP   !AF               !
  C5 32!                 !STAD  !                 ! WRITE BYTE COUNT TO CONTROL
  C6 00!                 ! ---- !adr 3200         !   BLOCK
  C7 32!                 ! ---- !                 !
  C8 22!                 !STPD  !HL               ! WRITE FIRST CHARACTER ADDRESS
  C9 01!                 ! ---- !adr 3201         !   TO CONTROL BLOCK
  CA 32!                 ! ---- !                 !
  CB 3E!                 !LDAI  !                 ! SELECT CH A WRITE REGISTER 1
  CC 01!                 ! ---- !01               !
  CD D3!                 !OPA   !                 !
  CE F5!                 ! ---- !F5               !
  CF 3A!                 !LDAD  !                 ! ENABLE XMT INTERRUPT
1DD0 79!                 ! ---- !adr 1D79         !
  D1 1D!                 ! ---- !                 !
  D2 F6!                 !ORAI  !                 !
  D3 02!                 ! ---- !02               !
  D4 D3!                 !OPA   !                 !
  D5 F5!                 ! ---- !F5               !
  D6 C3!                 !JP    !UNCONDITIONAL    ! WRITE FIRST CHARACTER TO CH A
  D7 90!                 ! ---- !(XMT BLK A)      !
  D8 1D!   ----------    ! ---- !                 !
  D9 FF!                 !      !                 !
  DA FF!                 !      !                 !
  DB FF!                 !      !                 !
  DC FF!                 !      !                 !
  DD FF!                 !      !                 !
  DE FF!                 !      !                 !
  DF FF!                 !      !                 !
1DE0 F5!(START XMT B)    !PSP   !AF               ! WAIT FOR END OF TRANSMISSION
  E1 CD!                 !JS    !UNCONDITIONAL    !   IN PROGRESS
  E2 00!                 ! ---- !(END XMT B?)     !
  E3 1E!                 ! ---- !                 !
  E4 F1!                 !PLP   !AF               !
  E5 32!                 !STAD  !                 ! WRITE BYTE COUNT TO CONTROL
  E6 03!                 ! ---- !adr 3203         !   BLOCK A
  E7 32!                 ! ---- !                 !
  E8 22!                 !STPD  !HL               ! WRITE FIRST CHARACTER ADDRESS
  E9 04!                 ! ---- !adr 3204         !   TO CONTROL BLOCK
  EA 32!                 ! ---- !                 !
  EB 3E!                 !LDAI  !                 ! SELECT CH B WRITE REGISTER 1
  EC 01!                 ! ---- !01               !
  ED D3!                 !OPA   !                 !
  EE F7!                 ! ---- !F7               !
  EF 3A!                 !LDAD  !                 ! ENABLE XMT INTERRUPT
1DF0 6B!                 ! ---- !adr 1D6B         !
  F1 1D!                 ! ---- !                 !
  F2 F6!                 !ORAI  !                 !
  F3 02!                 ! ---- !02               !
  F4 D3!                 !OPA   !                 !
  F5 F7!                 ! ---- !F7               !
  F6 C3!                 !JP    !UNCONDITIONAL    ! WRITE FIRST CHARACTER TO CH B
  F7 80!                 ! ---- !(XMT BLK B)      !
  F8 1D!   ----------    ! ---- !                 !
1DF9 0C!(XMT-OFF)        !ICR   !C                ! POINT TO CHANNEL CONTROL PORT
  FA 3E!                 !LDAI  !                 ! RESET PENDING XMT INTERRUPT
  FB 28!                 ! ---- !28               !
  FC ED!                 !OPAN  !(C)              !
  FD 79!                 ! ---- !                 !
  FE C9!   ----------    !RTS   !UNCONDITIONAL    !
  FF FF!                 !      !                 !
```

Figure 4-22. 7806 (Continued)

```
ADRdata!     LABEL       !INSTR !    MODIFIER     !          COMMENTS
-------!----------------!------!-----------------!---------------------------
1E00 3A!(END XMT B?)     !LDAD  !                 ! WAIT FOR LAST CHARACTER IN
  01 03!                 ! ---- !adr 3203         !   BLOCK TO BE WRITTEN TO
  02 32!                 ! ---- !                 !   CHANNEL B XMT
  03 FE!                 !CPAI  !                 !
  04 00!                 ! ---- !00               !
  05 C2!                 !JP    !NOT ZERO         !
  06 00!                 ! ---- !(END XMT B?)     !
  07 1E!                 ! ---- !                 !
1E08 DB!BUFF EMPTY*  B   !IPA   !                 ! WAIT FOR LAST CHARACTER TO
  09 F7!                 ! ---- !F7               !   CLEAR XMT BUFFER
  0A E6!                 !ANAI  !                 !
  0B 04!                 ! ---- !04               !
  0C CA!                 !JP    !ZERO             !
  0D 08!                 ! ---- !BUFF EMPTY*  B   !
  0E 1E!                 ! ---- !                 !
  0F C9!   ----------    !RTS   !UNCONDITIONAL    !
1E10 3A!(END XMT A?)     !LDAD  !                 ! WAIT FOR LAST CHARACTER IN
  11 00!                 ! ---- !adr 3200         !   BLOCK TO BE WRITTEN TO
  12 32!                 ! ---- !                 !   CHANNEL A XMT
  13 FE!                 !CPAI  !                 !
  14 00!                 ! ---- !00               !
  15 C2!                 !JP    !NOT ZERO         !
  16 10!                 ! ---- !(END XMT A?)     !
  17 1E!                 ! ---- !                 !
1E18 DB!BUFF EMPTY*  A   !IPA   !                 ! WAIT FOR LAST CHARACTER TO
  19 F5!                 ! ---- !F5               !   CLEAR XMT BUFFER
  1A E6!                 !ANAI  !                 !
  1B 04!                 ! ---- !04               !
  1C CA!                 !JP    !ZERO             !
  1D 18!                 ! ---- !BUFF EMPTY*  A   !
  1E 1E!                 ! ---- !                 !
  1F C9!   ----------    !RTS   !UNCONDITIONAL    !
1E20 FF!                 !      !                 !
  21 FF!                 !      !                 !
  22 FF!                 !      !                 !
  23 FF!                 !      !                 !
  24 FF!                 !      !                 !
  25 FF!                 !      !                 !
  26 FF!                 !      !                 !
  27 FF!                 !      !                 !
  28 FF!                 !      !                 !
  29 FF!                 !      !                 !
  2A FF!                 !      !                 !
  2B FF!                 !      !                 !
  2C FF!                 !      !                 !
  2D FF!                 !      !                 !
  2E FF!                 !      !                 !
  2F FF!                 !      !                 !
1E30 FF!                 !      !                 !
  31 FF!                 !      !                 !
  32 FF!                 !      !                 !
  33 FF!                 !      !                 !
  34 FF!                 !      !                 !
  35 FF!                 !      !                 !
  36 FF!                 !      !                 !
  37 FF!                 !      !                 !
  38 FF!                 !      !                 !
  39 FF!                 !      !                 !
  3A FF!                 !      !                 !
  3B FF!                 !      !                 !
  3C FF!                 !      !                 !
  3D FF!                 !      !                 !
  3E FF!                 !      !                 !
  3F FF!                 !      !                 !
```

Figure 4-22. 7806 (Continued)

```
ADRdata!     LABEL       !INSTR !    MODIFIER    !          COMMENTS
-------!---------------!------!----------------!---------------------------
1E40 FB!(RCV & ECHO A)  !ENI   !                ! ENABLE INTERRUPTS
  41 F5!                !PSP   !AF               ! SAVE PROCESSOR REGISTERS
  42 C5!                !PSP   !BC               !
  43 0E!                !LDRI  !C                ! RECEIVE CHARACTER AND TRANSMIT
  44 F4!                ! ---- !F4               !   IT BACK
  45 CD!                !JS    !UNCONDITIONAL    !
  46 60!                ! ---- !(RCV & ECHO)     !
  47 1E!                ! ---- !                 !
  48 CD!                !JS    !UNCONDITIONAL    ! FURTHER PROCESS THE RECEIVED
  49 --!                ! ---- !(PROCESS CHAR)   !   CHARACTER (USER SUPPLIED)
  4A --!                ! ---- !                 !
  4B C1!                !PLP   !BC               ! RESTORE PROCESSOR REGISTERS
  4C F1!                !PLP   !AF               !
  4D ED!                !RTI   !UNCONDITIONAL    ! RETURN
  4E 4D!    ----------  ! ---- !                 !
  4F FF!                !      !                 !
1E50 FB!(RCV & ECHO B)  !ENI   !                ! ENABLE INTERRUPTS
  51 F5!                !PSP   !AF               ! SAVE PROCESSOR REGISTERS
  52 C5!                !PSP   !BC               !
  53 0E!                !LDRI  !C                ! RECEIVE CHARACTER AND TRANSMIT
  54 F6!                ! ---- !F6               !   IT BACK
  55 CD!                !JS    !UNCONDITIONAL    !
  56 60!                ! ---- !(RCV & ECHO)     !
  57 1E!                ! ---- !                 !
  58 CD!                !JS    !UNCONDITIONAL    ! FURTHER PROCESS THE RECEIVED
  59 --!                ! ---- !(PROCESS CHAR)   !   CHARACTER (USER SUPPLIED)
  5A --!                ! ---- !                 !
  5B C1!                !PLP   !BC               ! RESTORE PROCESSOR REGISTERS
  5C F1!                !PLP   !AF               !
  5D ED!                !RTI   !UNCONDITIONAL    ! RETURN
  5E 4D!    ----------  ! ---- !                 !
  5F FF!                !      !                 !
1E60 ED!(RCV & ECHO)    !IPAN  !(C)              ! READ CHARACTER FROM DART
  61 78!                ! ---- !                 !   DATA PORT
  62 00!                !NOP   !                 ! JUMP TO SUBROUTINE "PROCESS
  63 00!                !NOP   !                 !   CHARACTER" MAY GO HERE
  64 00!                !NOP   !                 !
  65 ED!                !OPAN  !(C)              ! WRITE CHARACTER TO DART
  66 79!                ! ---- !                 !   DATA PORT
  67 C9!    ----------  !RTS   !UNCONDITIONAL    !
  68 FF!                !      !                 !
  69 FF!                !      !                 !
  6A FF!                !      !                 !
  6B FF!                !      !                 !
  6C FF!                !      !                 !
  6D FF!                !      !                 !
  6E FF!                !      !                 !
  6F FF!                !      !                 !
1E70 FF!                !      !                 !
  71 FF!                !      !                 !
  72 FF!                !      !                 !
  73 FF!                !      !                 !
  74 FF!                !      !                 !
  75 FF!                !      !                 !
  76 FF!                !      !                 !
  77 FF!                !      !                 !
  78 FF!                !      !                 !
  79 FF!                !      !                 !
  7A FF!                !      !                 !
  7B FF!                !      !                 !
  7C FF!                !      !                 !
  7D FF!                !      !                 !
  7E FF!                !      !                 !
  7F FF!                !      !                 !
```

Figure 4-22. 7806 (Continued)

```
ADRdata!      LABEL       !INSTR !    MODIFIER    !          COMMENTS
-------!----------------!------!----------------!-----------------------------
3200 FF!CH A CNTRL BLK   !       !                ! NUMBER OF CHARACTERS IN BLOCK
  01 FF!                 !       !                ! ADDRESS OF FIRST CHARACTER
  02 FF!                 !       !                !
3203 FF!CH B CNTRL BLK   !       !                ! NUMBER OF CHARACTERS IN BLOCK
  04 FF!                 !       !                ! ADDRESS OF FIRST CHARACTER
  05 FF!                 !       !                !
  06 FF!                 !       !                !
  07 FF!                 !       !                !
  08 FF!                 !       !                !
  09 FF!                 !       !                !
  0A FF!                 !       !                !
  0B FF!                 !       !                !
  0C FF!                 !       !                !
  0D FF!                 !       !                !
  0E FF!                 !       !                !
  0F FF!                 !       !                !
3210 FF!                 !       !                !
  11 FF!                 !       !                !
  12 FF!                 !       !                !
  13 FF!                 !       !                !
  14 FF!                 !       !                !
  15 FF!                 !       !                !
  16 FF!                 !       !                !
  17 FF!                 !       !                !
  18 FF!                 !       !                !
  19 FF!                 !       !                !
  1A FF!                 !       !                !
  1B FF!                 !       !                !
  1C FF!                 !       !                !
  1D FF!                 !       !                !
  1E FF!                 !       !                !
  1F FF!                 !       !                !
3220 FF!                 !       !                !
  21 FF!                 !       !                !
  22 FF!                 !       !                !
  23 FF!                 !       !                !
  24 FF!                 !       !                !
  25 FF!                 !       !                !
  26 FF!                 !       !                !
  27 FF!                 !       !                !
  28 FF!                 !       !                !
  29 FF!                 !       !                !
  2A FF!                 !       !                !
  2B FF!                 !       !                !
  2C FF!                 !       !                !
  2D FF!                 !       !                !
  2E FF!                 !       !                !
  2F FF!                 !       !                !
3230 FF!                 !       !                !
  31 FF!                 !       !                !
  32 FF!                 !       !                !
  33 FF!                 !       !                !
  34 FF!                 !       !                !
  35 FF!                 !       !                !
  36 FF!                 !       !                !
  37 FF!                 !       !                !
  38 FF!                 !       !                !
  39 FF!                 !       !                !
  3A FF!                 !       !                !
  3B FF!                 !       !                !
  3C FF!                 !       !                !
  3D FF!                 !       !                !
  3E FF!                 !       !                !
  3F FF!                 !       !                !
```

Figure 4-22. 7806 (Continued)

```
ADRdata!     LABEL       !INSTR !    MODIFIER    !          COMMENTS
-------!----------------!------!----------------!----------------------------
FF00 80!INTERRUPT VECTOR!  80  !                ! DART CH B SERVICE ROUTINES
FF01 1D!TABLE            !  1D  !                !   XMT BUFFER EMPTY
  02 FF!                 !      !                !   HANDSHAKE CHANGE
  03 FF!                 !      !                !
  04 50!                 !  50  !                !   RECEIVE CHARACTER AVAILABLE
  05 1E!                 !  1E  !                !
  06 FF!                 !      !                !   SPECIAL RECEIVE CONDITION
  07 FF!                 !      !                !
  08 90!                 !  90  !                ! DART CH A SERVICE ROUTINES
  09 1D!                 !  1D  !                !   XMT BUFFER EMPTY
  0A FF!                 !      !                !   HANDSHAKE CHANGE
  0B FF!                 !      !                !
  0C 40!                 !  40  !                !   RECEIVE CHARACTER AVAILABLE
  0D 1E!                 !  1E  !                !
  0E FF!                 !      !                !   SPECIAL RECEIVE CONDITION
  0F FF!                 !      !                !
FF10 FF!                 !      !                !
  11 FF!                 !      !                !
  12 FF!                 !      !                !
  13 FF!                 !      !                !
  14 FF!                 !      !                !
  15 FF!                 !      !                !
  16 FF!                 !      !                !
  17 FF!                 !      !                !
  18 FF!                 !      !                !
  19 FF!                 !      !                !
  1A FF!                 !      !                !
  1B FF!                 !      !                !
  1C FF!                 !      !                !
  1D FF!                 !      !                !
  1E FF!                 !      !                !
  1F FF!                 !      !                !
FF20 FF!                 !      !                !
  21 FF!                 !      !                !
  22 FF!                 !      !                !
  23 FF!                 !      !                !
  24 FF!                 !      !                !
  25 FF!                 !      !                !
  26 FF!                 !      !                !
  27 FF!                 !      !                !
  28 FF!                 !      !                !
  29 FF!                 !      !                !
  2A FF!                 !      !                !
  2B FF!                 !      !                !
  2C FF!                 !      !                !
  2D FF!                 !      !                !
  2E FF!                 !      !                !
  2F FF!                 !      !                !
FF30 FF!                 !      !                !
  31 FF!                 !      !                !
  32 FF!                 !      !                !
  33 FF!                 !      !                !
  34 FF!                 !      !                !
  35 FF!                 !      !                !
  36 FF!                 !      !                !
  37 FF!                 !      !                !
  38 FF!                 !      !                !
  39 FF!                 !      !                !
  3A FF!                 !      !                !
  3B FF!                 !      !                !
  3C FF!                 !      !                !
  3D FF!                 !      !                !
  3E FF!                 !      !                !
  3F FF!                 !      !                !
```

Figure 4-22. 7806 (Continued)

NAME	DESCRIPTION	MAIN SUB-ROUTINE	MAY BE USED BY ITSELF	NOT USEFUL BY ITSELF
(INIT SER COMM)	Initialize Serial Communications Interface (CH B)	X		
(INIT INTER)	Initialize Z80 Interrupts		X	
(INIT CTC 0)	Initialize CTC Channel 0		X	
(INIT PAR PORT)	Initialize Parallel (printer) port (CH A)	X		
(STROBE*)	Provide data strobe for printer interface (CH A)		X	
(INIT DART CH)	Initialize one channel of DART		X	
(ENABLE HNDSHK)	Enable recognition of printer port handshake transitions (CH A)		X	
(XMT BLK B)	Serially transmit a block of data (CH B)			X
(XMT CHAR)	Serially transmit a character (CH B)			X
(BUSY?)	Check status of "BUSY" line from printer (CH A)		X	
(PAPER END?)	Check status of "PE" line from printer (CH A)		X	
(STRT XMT B)	Start serial block transmission (CH B)	X		
(XMT OFF)	Turn off transmitter interrupt at end of block (CH B)			X
(END XMT B?)	Verify that entire block has been transmitted (CH B)			X
(PRINT DATA A)	Output a block of characters through printer interface (CH A)	X		
(RCV & ECHO B)	Serially receive a character and re-transmit it (CH B)	X		
(RCV & ECHO)	Receive and echo a character (CH B)			X
(PRINT DATA)	Output a block of characters through printer interface (CH A)			X

Figure 4-23. 7807 Subroutine Types

If the software provided is used as a package (accessing just the five main 7806 subroutines), only a few requirements must be met by the user. The five main subroutines and their requirements are given in figure 4-24. The following list details the requirements and the points to be considered before implementing the software.

Serial Port Considerations:

- The user may want to provide software for serial handshake control or for responding to changes in incoming serial handshaking.

- The software does not provide for response to errors in received data.

- The software does not store or respond to received data. The serial character is simply received and echoed. A Jump to Subroutine instruction is provided to jump to the user's own subroutine (PROCESS CHAR). The user's subroutine may then store and/or respond to the character according to the requirements of the application.

- The jump to (PROCESS CHAR) may ocur before or after the character is echoed (for before the echo, address hex 1E57; for after the echo, address hex 1E48).

- This software operates with Z80 interrupt mode 2. It selects mode 2 and enables the interrupt as part of the channel B initialization procedure. It also loads the Z80 interrupt register. The data to be loaded into the I register must be selected by the user (see address 1D1B).

- The baud rate and various other factors selected by the (INIT SER COMM) subroutine should be checked to see that they match the application. The initialized values are given in the description of the (INIT SER COMM) routine.

- The number of bytes to be transferred must be in register A (FF = 255) and the address of the data to be transferred in the HL register pair before using the transmit routines.

- Keep in mind the 7807 onboard interrupt priority (figure 3-6, page 3-13).

Parallel (Printer) Port Considerations:

- Channel A of the DART must be initialized before data can be output from the onboard parallel port. The (INIT PAR PORT) subroutine performs this function.

- The parallel port subroutines included in this manual are not interrupt driven. If desired, users can provide their own, more sophisticated interrupt-controlled print routines. The modules included in this manual provide a simple, useful means of operating the printer port and illustrate an example of DART register polling and bit manipulation.

- The PRINT subroutines included in this manual require a table in memory which contains the data to be printed, the table address (first byte location) in the HL register pair, and the number of characters to be printed (in hex) in the accumulator prior to execution. The (PRINT DATA A) subroutine performs some housekeeping directly if the user prefers to use an off-card port or perform housekeeping tailored to the application.

NAME	DESCRIPTION	USE REQUIREMENTS
(INIT SER COMM)	Initialize Serial Communications Interface (Entry Address 1D00)	Interrupt vector page address at 1D1B
(INIT PAR PORT)	Initialize parallel (printer) interface (Entry Address 1D30)	None
(STRT XMT B)	Transmit a block of data (CH B) (Entry Address 1DE0)	Data must be in memory, data pointer in register pair HL, and number of bytes in register A
(PRINT DATA A)	Outputs a block of characters through parallel printer interface (Entry Address 1E20)	Data must be in memory, data pointer in register pair HL, and number of bytes in register A
(RCV & ECHO B)	Receive and echo a character (CH B) (Entry Address 1E50)	User must provide software to store and/or respond to the received character

Figure 4-24. 7807 Main Subroutines Use Requirements

7807 SUBROUTINE SPECIFICATIONS: (INIT SER COMM)

ENTRY ADDRESS	:	1D00
MEMORY REQUIREMENTS		
Program Memory	:	18 (61 with subroutines and tables)
RAM	:	1
Stack	:	2
REGISTERS ALTERED	:	A, F, B, C, H, L, I
SUBROUTINES USED	:	(INIT CTC 0), (INIT DART CH), (INIT INTER)
EXECUTION TIME	:	595 Time States
ENTRY REQUIREMENTS	:	None
EXIT CONDITIONS	:	N/A
FUNCTIONAL DESCRIPTION	:	

This subroutine initializes channel B of the DART, channel 0 of the CTC (to generate a baud rate for the serial channel), and the Z80 interrupt. The initialized conditions should be checked to be sure they match the application. The initialized values are shown in figure 4-25. Refer to figures 3-4 and 3-10, pages 3-4 and 3-18, while reviewing these. The initial baud rate is 9600 using the 4 MHz version (7807-0).

ADDRESS	FUNCTION	DATA
1D21	CTC CH 0 Control Register	0100 0101
1D25	CTC CH 0 Load Register	0000 1101
1D63	DART CH B Control Register 2	0000 0000
1D65	DART CH B Control Register 4	0100 1111
1D67	DART CH B Control Register 5	1010 1010
1D69	DART CH B Control Register 3	0110 0001
1D6B	DART CH B Control Register 1	0001 0100

Figure 4-25. Serial Initialization Values (7807)

7807 SUBROUTINE SPECIFICATIONS: (INIT INTER)

ENTRY ADDRESS	:	1D18
MEMORY REQUIREMENTS		
Program Memory	:	8
RAM	:	0
Stack	:	0
REGISTERS ALTERED	:	A, I
SUBROUTINES USED	:	None
EXECUTION TIME	:	38 Time States
ENTRY REQUIREMENTS	:	None
EXIT CONDITIONS	:	N/A
FUNCTIONAL DESCRIPTION	:	

This subroutine selects Z80 interrupt mode 2 and enables the Z80 interrupt. It also loads the Z80 Interrupt register. The byte to be loaded in the Interrupt register must be provided by the user and stored at address 1D1B.

7807 SUBROUTINE SPECIFICATIONS: (INIT CTC 0)

ENTRY ADDRESS	:	1D20
MEMORY REQUIREMENTS		
Program Memory	:	9
RAM	:	0
Stack	:	0
REGISTERS ALTERED	:	A
SUBROUTINES USED	:	None
EXECUTION TIME	:	46 Time States
ENTRY REQUIREMENTS	:	None
EXIT CONDITIONS	:	N/A
FUNCTIONAL DESCRIPTION	:	

This subroutine initializes CTC channel 0. The Vector register is not loaded. Refer to figure 4-25 for the initialized values.

7807 SUBROUTINE SPECIFICATIONS: (INIT PAR PORT)

ENTRY ADDRESS	:	1D30
MEMORY REQUIREMENTS		
Program Memory	:	15 (42 with subroutines and table)
RAM	:	1
Stack	:	2
REGISTERS ALTERED	:	A, F, B, C, H, L
SUBROUTINES USED	:	(INIT DART CH), (ENABLE HNDSHK)
EXECUTION TIME	:	114 Time States
ENTRY REQUIREMENTS	:	None
EXIT CONDITIONS	:	N/A
FUNCTIONAL DESCRIPTION	:	

This module initializes channel A of the DART, stores an initial byte count of 00 in RAM at 3200 and enables recognition of channel A handshake line transitions. The initialized values are shown in figure 4-26. The initialized values should be checked to be sure they match the application. Refer to figure 3-4, page 3-4, while reviewing the initialized values.

ADDRESS	FUNCTION	DATA
1D73	DART CH A Control Register 4	0000 1100
1D75	DART CH A Control Register 5	0000 0000
1D77	DART CH A Control Register 3	0000 0000
1D79	DART CH A Control Register 1	0000 0000

Figure 4-26. Parallel Initialization Values (7807)

7807 SUBROUTINE SPECIFICATIONS: (INIT DART CH)

ENTRY ADDRESS	:	1D50
MEMORY REQUIREMENTS		
Program Memory	:	12 (plus 10-byte table for Channel A or 12-byte table for Channel B)
RAM	:	0
Stack	:	0
REGISTERS ALTERED	:	A, B, C, H, L, F
SUBROUTINES USED	:	None
EXECUTION TIME	:	420 Time States maximum
ENTRY REQUIREMENTS		
Register Pair HL	:	Points to Ch A data table (1D70) or Ch B data table (1D60)
EXIT CONDITIONS	:	N/A
FUNCTIONAL DESCRIPTION	:	

This subroutine initializes one channel of the DART. The channel is selected by loading register pair HL with the appropriate address (see Entry Requirements). Refer to figure 4-25, page 4-29, for the initialized values.

7807 SUBROUTINE SPECIFICATIONS: (STROBE*)

ENTRY ADDRESS	:	1D40
MEMORY REQUIREMENTS		
Program Memory	:	16
RAM	:	0
Stack	:	0
REGISTERS ALTERED	:	A
SUBROUTINES USED	:	None
EXECUTION TIME	:	79 Time States
ENTRY REQUIREMENTS	:	None
EXIT CONDITIONS	:	N/A
FUNCTIONAL DESCRIPTION	:	

This module provides a low-level active data strobe pulse as required by the Centronics printer interface convention. The STROBE* pulse generated is approximately 9 microseconds in duration, which far exceeds the requirements of most printers. What actually occurs is a momentary toggle of the DART channel A DTR handshake line, which is output to pin 1 of the parallel port connector J1.

7807 SUBROUTINE SPECIFICATIONS: (ENABLE HNDSHK)

ENTRY ADDRESS	:	1D7B
MEMORY REQUIREMENTS		
Program Memory	:	5
RAM	:	0
Stack	:	0
REGISTERS ALTERED	:	A
SUBROUTINES USED	:	None
EXECUTION TIME	:	28 Time States
ENTRY REQUIREMENTS	:	None
EXIT CONDITIONS	:	N/A
FUNCTION DESCRIPTION	:	

This subroutine executes a Reset Handshake Interrupt command to Channel A of the DART. It is advisable to execute this command immediately prior to reading the status of the incoming handshake lines. In the case of the printer routines, (ENABLE HNDSHK) is executed prior to reading the status of "BUSY" or "PAPER END." Refer to the DART register descriptions in section 3.

7807 SUBROUTINE SPECIFICATIONS: (BUSY?)

ENTRY ADDRESS	:	1DC0
MEMORY REQUIREMENTS		
Program Memory	:	13 (18 with subroutine)
RAM	:	0
Stack	:	3
REGISTERS ALTERED	:	None
SUBROUTINES USED	:	(ENABLE HNDSHK)
EXECUTION TIME	:	104 Time States (if not busy)
ENTRY REQUIREMENTS	:	None
EXIT CONDITIONS	:	N/A
FUNCTIONAL DESCRIPTION	:	

This module checks for a change in the status of the "BUSY" line from the Centronics printer interface. The processor loops in this routine if a change in status has occurred (printer is busy) until another change in status occurs (printer is not busy).

7807 SUBROUTINE SPECIFICATIONS: (PAPER END?)

ENTRY ADDRESS	:	1DD0
MEMORY REQUIREMENTS		
Program Memory	:	13 (18 with subroutine)
RAM	:	0
Stack	:	3
REGISTERS ALTERED	:	None
SUBROUTINES USED	:	(ENABLE HNDSHK)
EXECUTION TIME	:	104 Time States (if not out of paper)
ENTRY REQUIREMENTS	:	None
EXIT CONDITIONS	:	N/A
FUNCTIONAL DESCRIPTION	:	

This module checks for a change in the status of the "PE" line from the Centronics printer interface. The processor loops in this routine if a change in status has occurred (printer has run out of paper) until another change in status occurs (paper has been refilled).

7807 SUBROUTINE SPECIFICATIONS: (STRT XMT B)

ENTRY ADDRESS	:	1DE0
MEMORY REQUIREMENTS		
Program Memory	:	26 (93 with subroutines)
RAM	:	3
Stack	:	16
REGISTERS ALTERED	:	None
SUBROUTINES USED	:	(END XMT B?), (XMT BLK B), (XMT CHAR), (XMT OFF)
EXECUTION TIME	:	443 Time States (if transmission of previous block has been completed)
ENTRY REQUIREMENTS		
Register A	:	Number of bytes to be sent (FF=255)
Register Pair HL	:	Address of first byte to be sent
Memory	:	Data to be sent
EXIT CONDITIONS	:	Byte count decremented and stored at 3203, data pointer incremented and stored at 3204 and 3205
FUNCTIONAL DESCRIPTION	:	

This subroutine transmits a block of data of up to 255 bytes. The program first checks to see if a previous block is still in the process of being sent. If it is, the program stays in a continuous checking loop. When the block can be sent, the first byte is sent and the program exits. Thereafter, whenever the transmitter indicates it is ready, via interrupt, the program responds by sending another byte. When the transmitter interrupts after sending the last byte, the program turns off the transmitter interrupt.

7807 SUBROUTINE SPECIFICATIONS: (PRINT DATA A)

ENTRY ADDRESS	:	1E20
MEMORY REQUIREMENTS		
Program Memory	:	25 (150 with subroutines, plus table)
RAM	:	3
Stack	:	6
REGISTERS ALTERED	:	A, H, L
SUBROUTINES USED	:	(BUSY?), (PAPER END?), (PRINT DATA)
EXECUTION TIME	:	869 Time States (for A = 1 character)
ENTRY REQUIREMENTS		
Register A	:	Number of characters to be printed (FF=255)
Register Pair HL	:	Address of first character to be printed
Memory	:	Table of characters to be printed.
EXIT CONDITIONS	:	Character count decremented and stored at 3200, character pointer incremented and stored at 3201 and 3202.
FUNCTIONAL DESCRIPTION	:	

This subroutine outputs a block of data of up to 255 characters (such as ASCII bytes) through an on-card parallel port to a Centronics-type printer interface.

The program first checks to see if the printer is on line and able to receive characters. The program checks the status of the "BUSY" and "PAPER END" handshake lines after each character is output. The subroutine exits when the character count, which decrements after each output instruction, is zero. (INIT PAR PORT) or a similar initialization routine must be executed prior to (PRINT DATA A). This subroutine is not interrupt controlled.

7807 SUBROUTINE SPECIFICATIONS: (RCV & ECHO B)

ENTRY ADDRESS	:	1E50
MEMORY REQUIREMENTS		
Program Memory	:	15 (21 with subroutines)
RAM	:	0
Stack	:	6
REGISTERS ALTERED	:	None
SUBROUTINES USED	:	(RCV & ECHO), (PROCESS CHAR). (PROCESS CHAR) must be supplied by the user
EXECUTION TIME	:	149 Time States
ENTRY REQUIREMENTS	:	N/A
FUNCTIONAL DESCRIPTION	:	

This is an interrupt service routine. It receives a character and immediately echoes it back on the same channel. The user must supply a program (PROCESS CHAR) to interpret and/or store the received character. A Jump to Subroutine instruction may be placed at address 1E58 for this purpose. This will jump to the (PROCESS CHAR) subroutine after the character has been echoed. Alternatively, a Jump to Subroutine instruction may be placed at address 1E62, which is before the character is echoed.

The receiver is enabled at initialization and, since reception is interrupt driven, the (PROCESS CHAR) routine must flag the processor that a message has been received. Otherwise data reception is completely transparent to the processor.

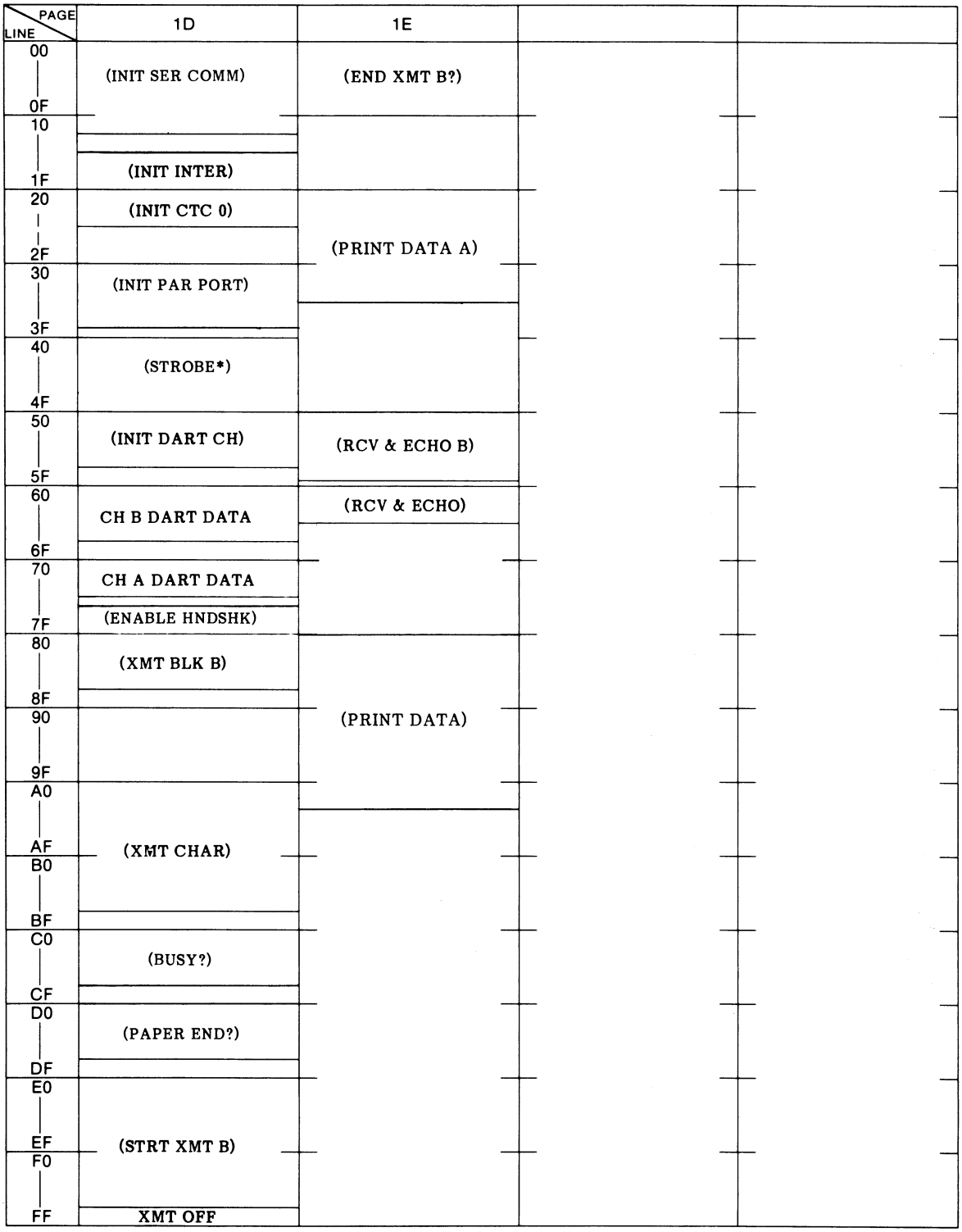


Figure 4-27. 1K Program Memory Map (7807)

PAGE ADDRESS: 32

LINE	LABEL
00	BYTE COUNT A
01	DATA LSB
02	POINTER A MSB
03	BYTE COUNT B
04	DATA LSB
05	POINTER B MSB
06	
07	
08	
09	
0A	
0B	
0C	
0D	
0E	
0F	
10	
11	
12	
13	
14	
15	
16	
17	
18	
19	
1A	
1B	
1C	
1D	
1E	
1F	
20	
21	
22	
23	
24	
25	
26	
27	
28	
29	
2A	
2B	
2C	
2D	
2E	
2F	
30	
31	
32	
33	
34	
35	
36	
37	
38	
39	
3A	
3B	
3C	
3D	
3E	
3F	

LINE	LABEL
40	
41	
42	
43	
44	
45	
46	
47	
48	
49	
4A	
4B	
4C	
4D	
4E	
4F	
50	
51	
52	
53	
54	
55	
56	
57	
58	
59	
5A	
5B	
5C	
5D	
5E	
5F	
60	
61	
62	
63	
64	
65	
66	
67	
68	
69	
6A	
6B	
6C	
6D	
6E	
6F	
70	
71	
72	
73	
74	
75	
76	
77	
78	
79	
7A	
7B	
7C	
7D	
7E	
7F	

LINE	LABEL
80	
81	
82	
83	
84	
85	
86	
87	
88	
89	
8A	
8B	
8C	
8D	
8E	
8F	
90	
91	
92	
93	
94	
95	
96	
97	
98	
99	
9A	
9B	
9C	
9D	
9E	
9F	
A0	
A1	
A2	
A3	
A4	
A5	
A6	
A7	
A8	
A9	
AA	
AB	
AC	
AD	
AE	
AF	
B0	
B1	
B2	
B3	
B4	
B5	
B6	
B7	
B8	
B9	
BA	
BB	
BC	
BD	
BE	
BF	

LINE	LABEL
C0	
C1	
C2	
C3	
C4	
C5	
C6	
C7	
C8	
C9	
CA	
CB	
CC	
CD	
CE	
CF	
D0	
D1	
D2	
D3	
D4	
D5	
D6	
D7	
D8	
D9	
DA	
DB	
DC	
DD	
DE	
DF	
E0	
E1	
E2	
E3	
E4	
E5	
E6	
E7	
E8	
E9	
EA	
EB	
EC	
ED	
EE	
EF	
F0	
F1	
F2	
F3	
F4	
F5	
F6	
F7	
F8	
F9	
FA	
FB	
FC	
FD	
FE	
FF	

Figure 4-28. 256-Byte RAM Memory Map (7807)

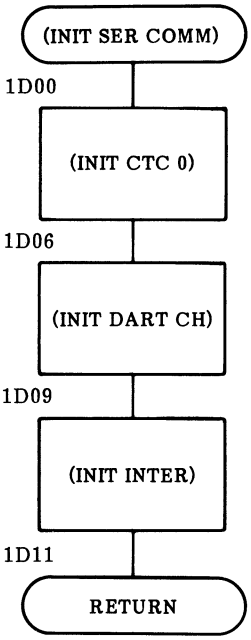


Figure 4-29. (INIT SER COMM) Flowchart (7807)

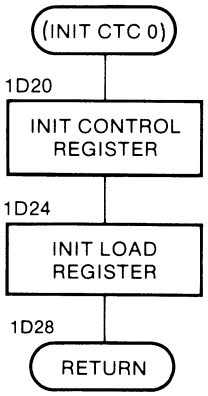


Figure 4-30. (INIT CTC 0) Flowchart (7807)

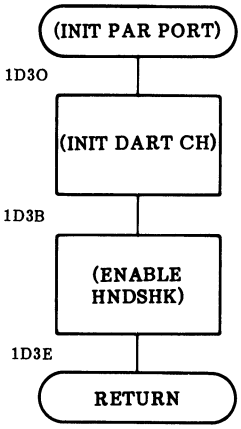


Figure 4-31. (INIT PAR PORT) Flowchart (7807)

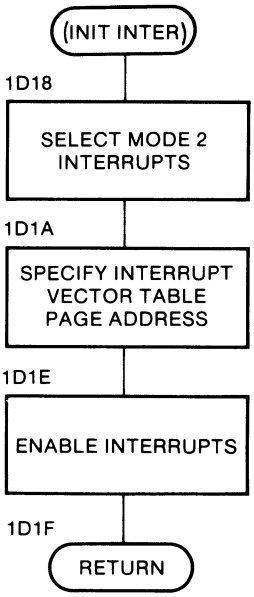


Figure 4-32. (INIT INTER) Flowchart (7807)

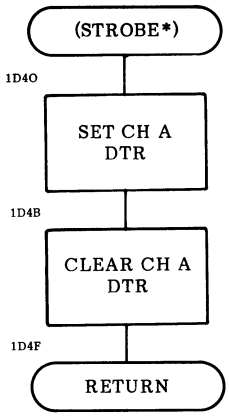


Figure 4-33. (STROBE*) Flowchart (7807)

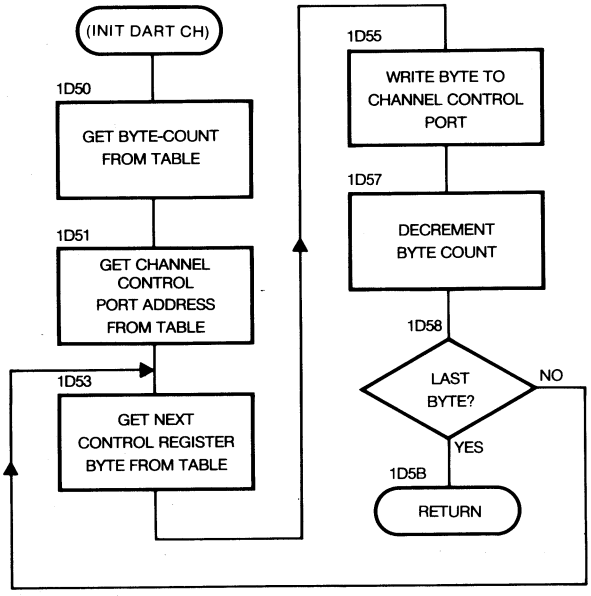


Figure 4-34. (INIT DART CH) Flowchart (7807)

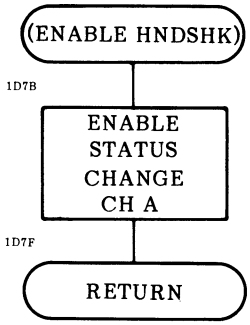


Figure 4-35. (ENABLE HNDSHK) Flowchart (7807)

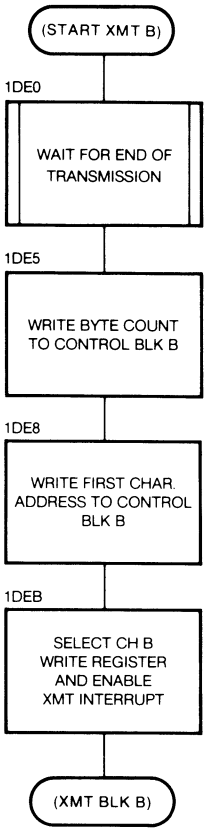


Figure 4-36. (STRT XMT B) Flowchart (7807)

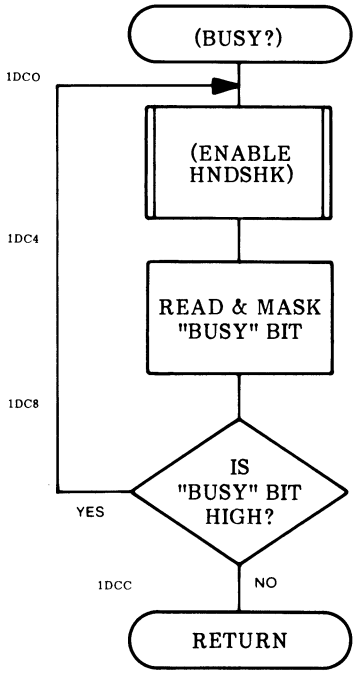


Figure 4-37. (BUSY?) Flowchart (7807)

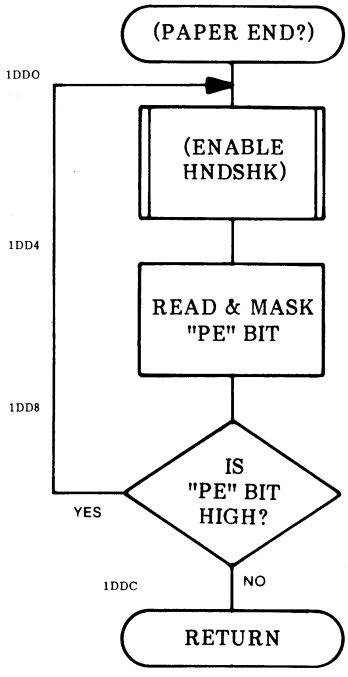


Figure 4-38. (PAPER END?) Flowchart (7807)

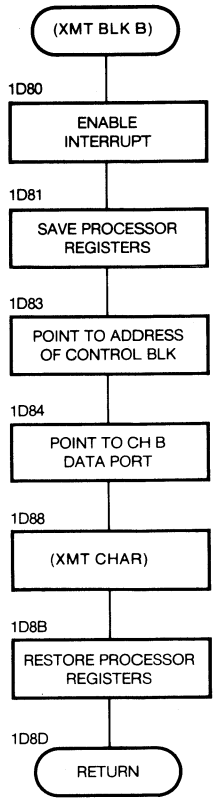


Figure 4-39. (XMT BLOCK B) Flowchart (7807)

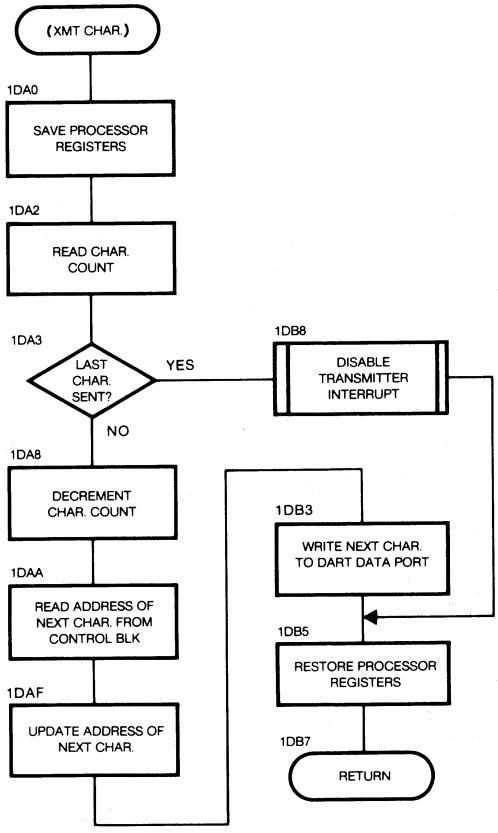


Figure 4-40. (XMT CHAR) Flowchart (7807)

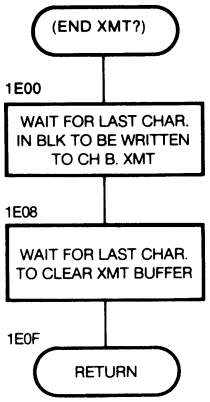


Figure 4-41. (END XMT?) Flowchart (7807)

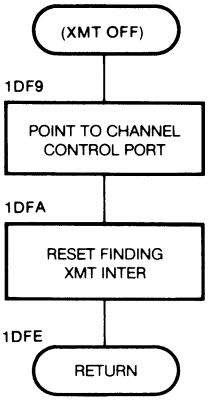


Figure 4-42. (XMT OFF) Flowchart (7807)

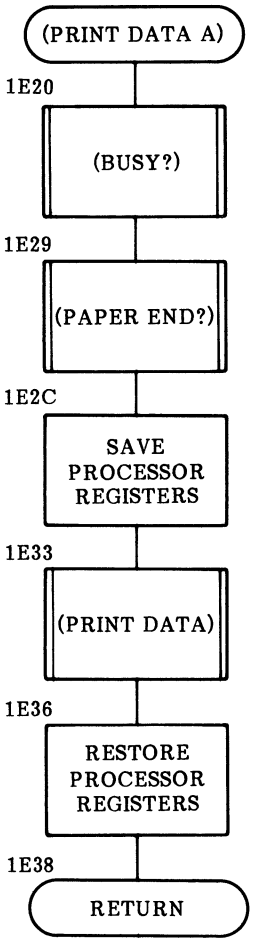


Figure 4-43. (PRINT DATA A) Flowchart (7807)

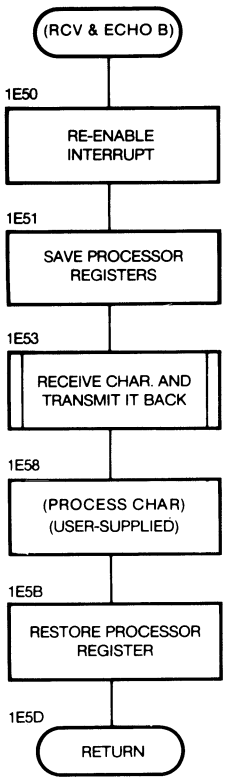


Figure 4-44. (RCV & ECHO B) Flowchart (7807)

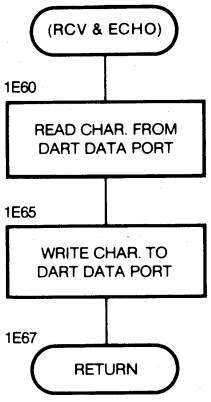


Figure 4-45. (RCV & ECHO) Flowchart (7807)

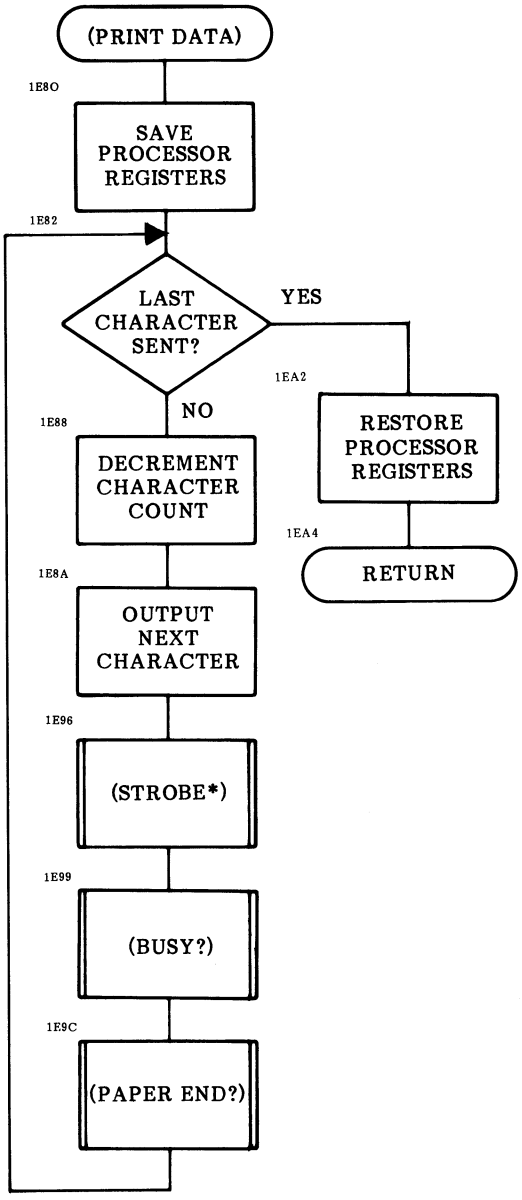


Figure 4-46. (PRINT DATA) Flowchart (7807)

```
ADRdata!     LABEL          !INSTR !    MODIFIER     !          COMMENTS
-------------------------------------------------------------------------------
1D00 CD!(INIT SER COMM)  !JS    !UN              !INITIALIZE CTC CH. 0
  01 20!                 ! ---- !(INIT CTC 0)     !  BAUD RATE 9600, CH. B
  02 1D!                 ! ---- !                 !
  03 21!                 !LDPI  !HL              !POINT TO CH. B DATA TABLE
  04 60!                 ! ---- !1D60 hex         !
  05 1D!                 ! ---- !                 !
  06 CD!                 !JS    !UN              !INITIALIZE DART, CH. B
  07 50!                 ! ---- !(INIT DART CH)  !
  08 1D!                 ! ---- !                 !
  09 CD!                 !JS    !UN              !INITIALIZE INTERRUPT STRUCTURE
  0A 18!                 ! ---- !(INIT INTER)    !
  0B 1D!                 ! ---- !                 !
  0C 3E!                 !LDAI  !                 !INITIALIZE DART CONTROL BLOCK
  0D 00!                 ! ---- !00 hex           !   COUNT AT 00
  0E 32!                 !STAD  !                 !
  0F 03!                 ! ---- !CH. B CNTRL BLK  !
1D10 32!                 ! ---- !                 !
  11 C9!   ----------    !RTS   !UN              !RETURN
  12 FF!                 !      !                 !
  13 FF!                 !      !                 !
  14 FF!                 !      !                 !
  15 FF!                 !      !                 !
  16 FF!                 !      !                 !
  17 FF!                 !      !                 !
1D18 ED!(INIT INTER)     !SEI   !02              !SELECT MODE 2 INTERRUPTS
  19 5E!                 ! ---- !                 !
  1A 3E!                 !LDAI  !                 !SPECIFY INTERRUPT VECTOR
  1B FF!                 ! ---- !FF hex           ! PAGE ADDRESS (USER SUPPLIED)
  1C ED!                 !LDRL  !I,A             !
  1D 47!                 ! ---- !                 !
  1E FB!                 !ENI   !                 !ENABLE INTERRUPTS
  1F C9!   ----------    !RTS   !UN              !RETURN
1D20 3E!(INIT CTC 0)     !LDAI  !                 !WRITE CHANNEL CONTROL WORD
  21 45!                 ! ---- !45 hex           !
  22 D3!                 !OPA   !                 !
  23 F0!                 ! ---- !F0 hex           !
  24 3E!                 !LDAI  !                 !WRITE TIME CONSTANT WORD
  25 0D!                 ! ---- !0D hex           !
  26 D3!                 !OPA   !                 !
  27 F0!                 ! ---- !F0 hex           !
  28 C9!   ----------    !RTS   !UN              !RETURN (SET FOR 9600 BAUD)
  29 FF!                 !      !                 !
  2A FF!                 !      !                 !
  2B FF!                 !      !                 !
  2C FF!                 !      !                 !
  2D FF!                 !      !                 !
  2E FF!                 !      !                 !
  2F FF!                 !      !                 !
1D30 21!(INIT PAR PORT)  !LDPI  !HL              !POINT TO CH. A DATA TABLE
  31 70!                 ! ---- !1D70 hex         !
  32 1D!                 ! ---- !                 !
  33 CD!                 !JS    !UN              !INITIALIZE DART, CH. A
  34 50!                 ! ---- !(INIT DART CH)  !
  35 1D!                 ! ---- !                 !
  36 3E!                 !LDAI  !                 !INITIALIZE DART CONTROL BLOCK
  37 00!                 ! ---- !00 hex           !  COUNT AT 00
  38 32!                 !STAD  !                 !
  39 00!                 ! ---- !CH. A CNTRL BLK  !
  3A 32!                 ! ---- !                 !
  3B CD!                 !JS    !UN              !ENABLE RECOGNITION OF HANDSHAKE
  3C 7B!                 ! ---- !(ENABLE HANDSHK)!  LINE TRANSITIONS, CH. A
  3D 1D!                 ! ---- !                 !
  3E C9!   ----------    !RTS   !UN              !RETURN
  3F FF!                 !      !                 !
```

Figure 4-47. Software Listing (7807)

```
ADRdata!     LABEL          !INSTR !    MODIFIER    !          COMMENTS
-------------------------------------------------------------------------------
1D40 3E!(STROBE*)           !LDAI  !                !STROBE* CH. A DTR LINE
  41 05!                    ! ---- !05 hex          !  POINT TO CH. A REG 5
  42 D3!                    !OPA   !                !
  43 F5!                    ! ---- !F5 hex          !
  44 3E!                    !LDAI  !                !SET CH. A DTR
  45 80!                    ! ---- !80 hex          !
  46 D3!                    !OPA   !                !
  47 F5!                    ! ---- !F5 hex          !
  48 3E!                    !LDAI  !                !POINT TO CH. A REG 5
  49 05!                    ! ---- !05 hex          !
  4A D3!                    !OPA   !                !
  4B F5!                    ! ---- !F5 hex          !
  4C AF!                    !XRA   !A               !CLEAR CH. A DTR
  4D D3!                    !OPA   !                !
  4E F5!                    ! ---- !F5 hex          !
  4F C9!   ----------       !RTS   !UN              !RETURN
1D50 46!(INIT DART CH)      !LDRN  !B,(HL)          !GET NUMBER OF BYTES TO OUTPUT
  51 23!                    !ICP   !HL              !GET CH. CONTROL PORT ADDRESS
  52 4E!                    !LDRN  !C,(HL)          !
1D53 23!NXT BYTE            !ICP   !HL              !OUTPUT TABLE TO CHANNEL CTRL
  54 7E!                    !LDAN  !(HL)            !  FIRST SELECTING WRITE REG
  55 ED!                    !OPAN  !(C)             !  AND THEN OUTPUTTING ITS
  56 79!                    ! ---- !                !  CONTROL BYTE
  57 05!                    !DCR   !B               !
  58 C2!                    !JP    !NOT ZERO        !
  59 53!                    ! ---- !NXT BYTE        !
  5A 1D!                    ! ---- !                !
  5B C9!   ----------       !RTS   !UN              !RETURN
  5C FF!                    !      !                !
  5D FF!                    !      !                !
  5E FF!                    !      !                !
  5F FF!                    !      !                !
1D60 0A!CH. B DATA          !  0A  !                !NUMBER OF BYTES TO OUTPUT
  61 F7!                    !  F7  !                !CHANNEL B CONTROL PORT
  62 02!                    !  02  !                !WRITE REG 2
  63 00!                    !  00  !                !CONTROL BYTE REG 2
  64 04!                    !  04  !                !WRITE REG 4
  65 4F!                    !  4F  !                !CONTROL BYTE REG 4
  66 05!                    !  05  !                !WRITE REG 5
  67 AA!                    !  AA  !                !CONTROL BYTE REG 5
  68 03!                    !  03  !                !WRITE REG 3
  69 61!                    !  61  !                !CONTROL BYTE REG 3
  6A 01!                    !  01  !                !WRITE REG 1
1D6B 14!adr 1D6B            !  14  !                !CONTROL BYTE REG 1
  6C FF!                    !      !                !
  6D FF!                    !      !                !
  6E FF!                    !      !                !
  6F FF!                    !      !                !
1D70 08!CH. A DATA          !  08  !                !NUMBER OF BYTES TO OUTPUT
  71 F5!                    !  F5  !                !CHANNEL A CONTROL PORT
  72 04!                    !  04  !                !WRITE REG 4
  73 0C!                    !  0C  !                !CONTROL BYTE REG 4
  74 05!                    !  05  !                !WRITE REG 5
  75 00!                    !  00  !                !CONTROL BYTE REG 5
  76 03!                    !  03  !                !WRITE REG 3
  77 00!                    !  00  !                !CONTROL BYTE REG 3
  78 01!                    !  01  !                !WRITE REG 1
  79 00!                    !  00  !                !CONTROL BYTE REG 1
  7A FF!                    !      !                !
1D7B 3E!(ENABLE HANDSHK)    !LDAI  !                !ENABLE STATUS CHANGE, CH. A
  7C 10!                    ! ---- !10 hex          !
  7D D3!                    !OPA   !                !
  7E F5!                    ! ---- !F5 hex          !
  7F C9!   ----------       !RTS   !UN              !RETURN
```

Figure 4-47. 7807 (Continued)

```
ADRdata!      LABEL       !INSTR !    MODIFIER    !          COMMENTS
-------------------------------------------------------------------------------
1D80 FB!(XMT BLK B)       !ENI   !                !ENABLE INTERRUPTS
  81 C5!                  !PSP   !BC              !SAVE PROCESSOR REGISTERS
  82 E5!                  !PSP   !HL              !
  83 21!                  !LDPI  !HL              !POINT TO ADDRESS OF CONTROL
  84 03!                  ! ---- !3203 hex        !  BLOCK
  85 32!                  ! ---- !                !
  86 0E!                  !LDRI  !C               !POINT TO CH. B DATA PORT
  87 F6!                  ! ---- !F6 hex          !
  88 CD!                  !JS    !UN              !TRANSMIT CHARACTER
  89 A0!                  ! ---- !(XMT CHAR)      !
  8A 1D!                  ! ---- !                !
  8B E1!                  !PLP   !HL              !RESTORE PROCESSOR REGISTERS
  8C C1!                  !PLP   !BC              !
  8D ED!                  !RTI   !UN              !RETURN FROM INTERRUPT
  8E 4D!                  ! ---- !                !
  8F FF!                  !      !                !
1D90 FF!                  !      !                !
  91 FF!                  !      !                !
  92 FF!                  !      !                !
  93 FF!                  !      !                !
  94 FF!                  !      !                !
  95 FF!                  !      !                !
  96 FF!                  !      !                !
  97 FF!                  !      !                !
  98 FF!                  !      !                !
  99 FF!                  !      !                !
  9A FF!                  !      !                !
  9B FF!                  !      !                !
  9C FF!                  !      !                !
  9D FF!                  !      !                !
  9E FF!                  !      !                !
  9F FF!                  !      !                !
1DA0 F5!(XMT CHAR)        !PSP   !AF              !SAVE PROCESSOR REGISTERS
  A1 D5!                  !PSP   !DE              !
  A2 7E!                  !LDAN  !(HL)            !HAS LAST CHARACTER BEEN SENT?
  A3 FE!                  !CPAI  !                !
  A4 00!                  ! ---- !00 hex          !
  A5 CA!                  !JP    !ZERO            !
  A6 B8!                  ! ---- !LAST B          !
  A7 1D!                  ! ---- !                !
  A8 3D!                  !DCA   !                !NO, DECREMENT COUNT AND STORE
  A9 77!                  !STAN  !(HL)            !
  AA 23!                  !ICP   !HL              !READ ADDRESS OF NEXT CHARACTER
  AB 5E!                  !LDRN  !E,(HL)          !   FROM CONTROL BLOCK
  AC 23!                  !ICP   !HL              !
  AD 56!                  !LDRN  !D,(HL)          !
  AE 1A!                  !LDAN  !(DE)            !READ NXT CHARACTER
  AF 13!                  !ICP   !DE              !UPDATE CHARACTER ADDRESS AND
1DB0 72!                  !STRN  !D,(HL)          !   STORE IN CNTRL BLOCK
  B1 2B!                  !DCP   !HL              !
  B2 73!                  !STRN  !E,(HL)          !
  B3 ED!                  !OPAN  !(C)             !WRITE TO CH. B DATA PORT
  B4 79!                  ! ---- !                !
1DB5 D1!RESTORE           !PLP   !DE              !RESTORE PROCESSOR REGISTERS
  B6 F1!                  !PLP   !AF              !
  B7 C9!   ----------     !RTS   !UN              !RETURN
1DB8 CD!LAST B            !JS    !UN              !LAST CHARACTER SENT, TURN OFF
  B9 F9!                  ! ---- !(XMT OFF)       !   XMT INTERRUPT
  BA 1D!                  ! ---- !                !
  BB C3!                  !JP    !UN              !GO RESTORE PROCESSOR REGISTERS
  BC B5!                  ! ---- !RESTORE         !
  BD 1D!   ----------     ! ---- !                !
  BE FF!                  !      !                !
  BF FF!                  !      !                !
```

Figure 4-47. 7807 (Continued)

```
ADRdata!     LABEL       !INSTR !    MODIFIER    !          COMMENTS
-------------------------------------------------------------------------------
1DC0 F5!(BUSY?)          !PSP   !AF              !SAVE PROCESSOR REGISTERS
1DC1 CD!AGAIN (BUSY)     !JS    !UN              !ENABLE REGOGNITION OF HAND-
  C2 7B!                 ! ---- !(ENABLE HANDSHK)!  SHAKE LINE TRANSITIONS, CH A.
  C3 1D!                 ! ---- !
  C4 DB!                 !IPA   !                !CHECK STATUS OF "BUSY" LINE
  C5 F5!                 ! ---- !F5 hex          !
  C6 E6!                 !ANAI  !                !
  C7 08!                 ! ---- !08 hex          !
  C8 CA!                 !JP    !ZERO            !IF BUSY, CHECK AGAIN
  C9 C1!                 ! ---- !AGAIN (BUSY)    !
  CA 1D!                 ! ---- !                !
  CB F1!                 !PLP   !AF              !IF NOT BUSY, RESTORE PROCESSOR
  CC C9!   ----------    !RTS   !UN              !   REGISTERS AND RETURN
  CD FF!                 !      !                !
  CE FF!                 !      !                !
  CF FF!                 !      !                !
1DD0 F5!(PAPER END?)     !PSP   !AF              !SAVE PROCESSOR REGISTERS
1DD1 CD!AGAIN (PE)       !JS    !UN              !ENABLE RECOGNITION OF HAND-
  D2 7B!                 ! ---- !(ENABLE HANDSHK)!   SHAKE LINE TRANSITIONS, CH.A
  D3 1D!                 ! ---- !                !
  D4 DB!                 !IPA   !                !CHECK STATUS OF "PE" LINE
  D5 F5!                 ! ---- !F5 hex          !
  D6 E6!                 !ANAI  !                !
  D7 20!                 ! ---- !20 hex          !
  D8 CA!                 !JP    !ZERO            !IF PAPER END, CHECK AGAIN
  D9 D1!                 ! ---- !AGAIN (PE)      !
  DA 1D!                 ! ---- !                !
  DB F1!                 !PLP   !AF              !IF NOT PAPER END, RESTORE
  DC C9!   ----------    !RTS   !UN              !  PROCESSOR REGISTERS AND RETRN
  DD FF!                 !      !                !
  DE FF!                 !      !                !
  DF FF!                 !      !                !
1DE0 F5!(START XMT B)    !PSP   !AF              !WAIT FOR END OF TRANSMISSION
  E1 CD!                 !JS    !UN              !   IN PROGRESS
  E2 00!                 ! ---- !(END XMT B?)    !
  E3 1E!                 ! ---- !                !
  E4 F1!                 !PLP   !AF              !
  E5 32!                 !STAD  !                !WRITE BYTE COUNT TO CONTROL
  E6 03!                 ! ---- !CH. B CNTRL BLK !   BLOCK B
  E7 32!                 ! ---- !                !
  E8 22!                 !STPD  !HL              !WRITE FIRST CHARACTER ADDRESS
  E9 04!                 ! ---- !adr 3204        !   TO CONTROL BLOCK B
  EA 32!                 ! ---- !                !
  EB 3E!                 !LDAI  !                !SELECT CH. B WRITE REGISTER 1
  EC 01!                 ! ---- !01 hex          !
  ED D3!                 !OPA   !                !
  EE F7!                 ! ---- !F7 hex          !
  EF 3A!                 !LDAD  !                !ENABLE XMT INTERRUPT
1DF0 6B!                 ! ---- !adr 1D6B        !
  F1 1D!                 ! ---- !                !
  F2 F6!                 !ORAI  !                !
  F3 02!                 ! ---- !02 hex          !
  F4 D3!                 !OPA   !                !
  F5 F7!                 ! ---- !F7 hex          !
  F6 C3!                 !JP    !UN              !WRITE FIRST CHARACTER TO CH. B
  F7 80!                 ! ---- !(XMT BLK B)     !
  F8 1D!   ----------    ! ---- !                !
1DF9 0C!(XMT OFF)        !ICR   !C               !POINT TO CHANNEL CONTROL PORT
  FA 3E!                 !LDAI  !                !RESET PENDING XMT INTERRUPT
  FB 28!                 ! ---- !28 hex          !
  FC ED!                 !OPAN  !(C)             !
  FD 79!                 ! ---- !                !
  FE C9!   ----------    !RTS   !UN              !RETURN
  FF FF!                 !      !                !
```

Figure 4-47. 7807 (Continued)

```
ADRdata!      LABEL       !INSTR !    MODIFIER    !           COMMENTS
-------------------------------------------------------------------------------
1E00 3A!(END XMT B?)      !LDAD  !                !WAIT FOR LAST CHARACTER IN
  01 03!                  ! ---- !CH. B CNTRL BLK !   BLOCK TO BE WRITTEN TO
  02 32!                  ! ---- !                !   CHANNEL B XMT
  03 FE!                  !CPAI  !                !
  04 00!                  ! ---- !00 hex          !
  05 C2!                  !JP    !NOT ZERO        !
  06 00!                  ! ---- !(END XMT B?)    !
  07 1E!                  ! ---- !                !
1E08 DB!BUFF EMPTY* B     !IPA   !                !WAIT FOR LAST CHARACTER TO
  09 F7!                  ! ---- !F7 hex          !   CLEAR XMT BUFER
  0A E6!                  !ANAI  !                !
  0B 04!                  ! ---- !04 hex          !
  0C CA!                  !JP    !ZERO            !
  0D 08!                  ! ---- !BUFF EMPTY* B   !
  0E 1E!                  ! ---- !                !
  0F C9!   ----------     !RTS   !UN              !RETURN
1E10 FF!                  !      !                !
  11 FF!                  !      !                !
  12 FF!                  !      !                !
  13 FF!                  !      !                !
  14 FF!                  !      !                !
  15 FF!                  !      !                !
  16 FF!                  !      !                !
  17 FF!                  !      !                !
  18 FF!                  !      !                !
  19 FF!                  !      !                !
  1A FF!                  !      !                !
  1B FF!                  !      !                !
  1C FF!                  !      !                !
  1D FF!                  !      !                !
  1E FF!                  !      !                !
  1F FF!                  !      !                !
1E20 32!(PRINT DATA A)    !STAD  !                !WRITE BYTE COUNT TO CONTROL
  21 00!                  ! ---- !CH. A CNTRL BLK !   BLOCK A
  22 32!                  ! ---- !                !
  23 22!                  !STPD  !HL              !WRITE FIRST CHARACTER ADDRESS
  24 01!                  ! ---- !adr 3201        !   TO CONTROL BLOCK A
  25 32!                  ! ---- !                !
  26 CD!                  !JS    !UN              !CHECK FOR BUSY
  27 C0!                  ! ---- !(BUSY?)         !
  28 1D!                  ! ---- !                !
  29 CD!                  !JS    !UN              !CHECK FOR PAPER
  2A D0!                  ! ---- !(PAPER END?)    !
  2B 1D!                  ! ---- !                !
  2C C5!                  !PSP   !BC              !SAVE PROCESSOR REGISTERS
  2D E5!                  !PSP   !HL              !
  2E 21!                  !LDPI  !HL              !POINT TO ADDRESS OF CONTROL
  2F 00!                  ! ---- !3200 hex        !   BLOCK
1E30 32!                  ! ---- !                !
  31 0E!                  !LDRI  !C               !POINT TO ON-BOARD PARALLEL PORT
  32 FD!                  ! ---- !FD hex          !
  33 CD!                  !JS    !UN              !PRINT IT
  34 80!                  ! ---- !(PRINT DATA)    !
  35 1E!                  ! ---- !                !
  36 E1!                  !PLP   !HL              !RESTORE PROCESSOR REGISTERS
  37 C1!                  !PLP   !BC              !
  38 C9!   ----------     !RTS   !UN              !RETURN
  39 FF!                  !      !                !
  3A FF!                  !      !                !
  3B FF!                  !      !                !
  3C FF!                  !      !                !
  3D FF!                  !      !                !
  3E FF!                  !      !                !
  3F FF!                  !      !                !
```

Figure 4-47. 7807 (Continued)

```
ADRdata!      LABEL        !INSTR !    MODIFIER    !          COMMENTS
-------------------------------------------------------------------------------
1E40 FF!                   !      !                !
  41 FF!                   !      !                !
  42 FF!                   !      !                !
  43 FF!                   !      !                !
  44 FF!                   !      !                !
  45 FF!                   !      !                !
  46 FF!                   !      !                !
  47 FF!                   !      !                !
  48 FF!                   !      !                !
  49 FF!                   !      !                !
  4A FF!                   !      !                !
  4B FF!                   !      !                !
  4C FF!                   !      !                !
  4D FF!                   !      !                !
  4E FF!                   !      !                !
  4F FF!                   !      !                !
1E50 FB!(RCV & ECHO B)     !ENI   !                !ENABLE INTERRUPTS
  51 F5!                   !PSP   !AF              !SAVE PROCESSOR REGISTERS
  52 C5!                   !PSP   !BC              !
  53 0E!                   !LDRI  !C               !RECEIVE CHARACTER AND TRANSMIT
  54 F6!                   ! ---- !F6 hex          !   IT BACK
  55 CD!                   !JS    !UN              !
  56 60!                   ! ---- !(RCV & ECHO)    !
  57 1E!                   ! ---- !                !
  58 00!                   !NOP   !                !FURTHER PROCESS THE RECEIVED
  59 00!                   !NOP   !                !   CHARACTER (USER-SUPPLIED)
  5A 00!                   !NOP   !                !
  5B C1!                   !PLP   !BC              !RESTORE PROCESSOR REGISTERS
  5C F1!                   !PLP   !AF              !
  5D ED!                   !RTI   !UN              !RETURN FROM INTERRUPT
  5E 4D!                   ! ---- !                !
  5F FF!                   !      !                !
1E60 ED!(RCV & ECHO)       !IPAN  !(C)             !READ CHARACTER FROM DART
  61 78!                   ! ---- !                !   DATA PORT
  62 00!                   !NOP   !                !JUMP TO SUBROUTINE "PROCESS
  63 00!                   !NOP   !                !   CHARACTER" MAY GO HERE
  64 00!                   !NOP   !                !
  65 ED!                   !OPAN  !(C)             !WRITE CHARACTER TO DART
  66 79!                   ! ---- !                !   DATA PORT
  67 C9!   ----------      !RTS   !UN              !RETURN
  68 FF!                   !      !                !
  69 FF!                   !      !                !
  6A FF!                   !      !                !
  6B FF!                   !      !                !
  6C FF!                   !      !                !
  6D FF!                   !      !                !
  6E FF!                   !      !                !
  6F FF!                   !      !                !
1E70 FF!                   !      !                !
  71 FF!                   !      !                !
  72 FF!                   !      !                !
  73 FF!                   !      !                !
  74 FF!                   !      !                !
  75 FF!                   !      !                !
  76 FF!                   !      !                !
  77 FF!                   !      !                !
  78 FF!                   !      !                !
  79 FF!                   !      !                !
  7A FF!                   !      !                !
  7B FF!                   !      !                !
  7C FF!                   !      !                !
  7D FF!                   !      !                !
  7E FF!                   !      !                !
  7F FF!                   !      !                !
```

Figure 4-47. 7807 (Continued)

```
ADRdata!      LABEL       !INSTR !     MODIFIER     !            COMMENTS
-------------------------------------------------------------------------------
1E80 F5!(PRINT DATA)      !PSP   !AF                !SAVE PROCESSOR REGISTERS
  81 D5!                  !PSP   !DE                !
1E82 7E!LAST?             !LDAN  !(HL)              !HAS LAST CHARACTER BEEN SENT?
  83 FE!                  !CPAI  !                  !
  84 00!                  ! ---- !00 hex            !
  85 CA!                  !JP    !ZERO              !
  86 A2!                  ! ---- !LAST A            !
  87 1E!                  ! ---- !                  !
  88 3D!                  !DCA   !                  !NO, DECREMENT COUNT AND STORE
  89 77!                  !STAN  !(HL)              !
  8A 23!                  !ICP   !HL                !READ ADDRESS OF NEXT CHARACTER
  8B 5E!                  !LDRN  !E,(HL)            !   FROM CONTROL BLOCK
  8C 23!                  !ICP   !HL                !
  8D 56!                  !LDRN  !D,(HL)            !
  8E 1A!                  !LDAN  !(DE)              !READ NEXT CHARACTER
  8F 13!                  !ICP   !DE                !UPDATE CHARACTER ADDRESS AND
1E90 72!                  !STRN  !D,(HL)            !   STORE IN CONTROL BLOCK
  91 2B!                  !DCP   !HL                !
  92 73!                  !STRN  !E,(HL)            !
  93 2B!                  !DCP   !HL                !POINT TO CHARACTER COUNT
  94 ED!                  !OPAN  !(C)               !OUTPUT CHARACTER TO PARALLEL
  95 79!                  ! ---- !                  !   PORT
  96 CD!                  !JS    !UN                !STROBE DATA
  97 40!                  ! ---- !(STROBE*)         !
  98 1D!                  ! ---- !                  !
  99 CD!                  !JS    !UN                !CHECK FOR BUSY
  9A C0!                  ! ---- !(BUSY?)           !
  9B 1D!                  ! ---- !                  !
  9C CD!                  !JS    !UN                !CHECK FOR PAPER
  9D D0!                  ! ---- !(PAPER END?)      !
  9E 1D!                  ! ---- !                  !
  9F C3!                  !JP    !UN                !CHECK FOR LAST CHARACTER
1EA0 82!                  ! ---- !LAST?             !
  A1 1E!  ----------      ! ---- !                  !
1EA2 D1!LAST A            !PLP   !DE                !RESTORE PROCESSOR REGISTERS
  A3 F1!                  !PLP   !AF                !
  A4 C9!  ----------      !RTS   !UN                !RETURN
  A5 FF!                  !      !                  !
  A6 FF!                  !      !                  !
  A7 FF!                  !      !                  !
  A8 FF!                  !      !                  !
  A9 FF!                  !      !                  !
  AA FF!                  !      !                  !
  AB FF!                  !      !                  !
  AC FF!                  !      !                  !
  AD FF!                  !      !                  !
  AE FF!                  !      !                  !
  AF FF!                  !      !                  !
1EB0 FF!                  !      !                  !
  B1 FF!                  !      !                  !
  B2 FF!                  !      !                  !
  B3 FF!                  !      !                  !
  B4 FF!                  !      !                  !
  B5 FF!                  !      !                  !
  B6 FF!                  !      !                  !
  B7 FF!                  !      !                  !
  B8 FF!                  !      !                  !
  B9 FF!                  !      !                  !
  BA FF!                  !      !                  !
  BB FF!                  !      !                  !
  BC FF!                  !      !                  !
  BD FF!                  !      !                  !
  BE FF!                  !      !                  !
  BF FF!                  !      !                  !
```

Figure 4-47. 7807 (Continued)

ADRdata	LABEL	INSTR	MODIFIER	COMMENTS
3200 FF	CH. A CNTRL BLK			NUMBER OF CHARACTERS IN BLOCK
3201 FF	adr 3201			ADDRESS OF FIRST CHARACTER
02 FF				
3203 FF	CH. B CNTRL BLK			NUMBER OF CHARACTERS IN BLOCK
3204 FF	adr 3204			ADDRESS OF FIRST CHARACTER
05 FF				
06 FF				
07 FF				
08 FF				
09 FF				
0A FF				
0B FF				
0C FF				
0D FF				
0E FF				
0F FF				
3210 FF				
11 FF				
12 FF				
13 FF				
14 FF				
15 FF				
16 FF				
17 FF				
18 FF				
19 FF				
1A FF				
1B FF				
1C FF				
1D FF				
1E FF				
1F FF				
3220 FF				
21 FF				
22 FF				
23 FF				
24 FF				
25 FF				
26 FF				
27 FF				
28 FF				
29 FF				
2A FF				
2B FF				
2C FF				
2D FF				
2E FF				
2F FF				
3230 FF				
31 FF				
32 FF				
33 FF				
34 FF				
35 FF				
36 FF				
37 FF				
38 FF				
39 FF				
3A FF				
3B FF				
3C FF				
3D FF				
3E FF				
3F FF				

Figure 4-47. 7807 (Continued)

```
ADRdata!      LABEL        !INSTR !     MODIFIER     !          COMMENTS
-------------------------------------------------------------------------------
FF00 80!INTERRUPT VECTOR!   80  !                  !DART CH. B SERVICE ROUTINES
FF01 1D!TABLE            !  1D  !                  !   XMT BUFFER EMPTY
  02 FF!                 !  FF  !                  !   HANDSHAKE CHANGE
  03 FF!                 !  FF  !                  !
  04 50!                 !  50  !                  !   RECEIVE CHARACTER AVAILABLE
  05 1E!                 !  1E  !                  !
  06 FF!                 !  FF  !                  !   SPECIAL RECEIVE CONDITION
  07 FF!                 !  FF  !                  !
  08 FF!                 !      !                  !
  09 FF!                 !      !                  !
  0A FF!                 !      !                  !
  0B FF!                 !      !                  !
  0C FF!                 !      !                  !
  0D FF!                 !      !                  !
  0E FF!                 !      !                  !
  0F FF!                 !      !                  !
FF10 FF!                 !      !                  !
  11 FF!                 !      !                  !
  12 FF!                 !      !                  !
  13 FF!                 !      !                  !
  14 FF!                 !      !                  !
  15 FF!                 !      !                  !
  16 FF!                 !      !                  !
  17 FF!                 !      !                  !
  18 FF!                 !      !                  !
  19 FF!                 !      !                  !
  1A FF!                 !      !                  !
  1B FF!                 !      !                  !
  1C FF!                 !      !                  !
  1D FF!                 !      !                  !
  1E FF!                 !      !                  !
  1F FF!                 !      !                  !
FF20 FF!                 !      !                  !
  21 FF!                 !      !                  !
  22 FF!                 !      !                  !
  23 FF!                 !      !                  !
  24 FF!                 !      !                  !
  25 FF!                 !      !                  !
  26 FF!                 !      !                  !
  27 FF!                 !      !                  !
  28 FF!                 !      !                  !
  29 FF!                 !      !                  !
  2A FF!                 !      !                  !
  2B FF!                 !      !                  !
  2C FF!                 !      !                  !
  2D FF!                 !      !                  !
  2E FF!                 !      !                  !
  2F FF!                 !      !                  !
FF30 FF!                 !      !                  !
  31 FF!                 !      !                  !
  32 FF!                 !      !                  !
  33 FF!                 !      !                  !
  34 FF!                 !      !                  !
  35 FF!                 !      !                  !
  36 FF!                 !      !                  !
  37 FF!                 !      !                  !
  38 FF!                 !      !                  !
  39 FF!                 !      !                  !
  3A FF!                 !      !                  !
  3B FF!                 !      !                  !
  3C FF!                 !      !                  !
  3D FF!                 !      !                  !
  3E FF!                 !      !                  !
  3F FF!                 !      !                  !
```

Figure 4-47. 7807 (Continued)

REFERENCE DRAWINGS

The schematics (figure 5-1 for the 7806 and figure 5-3 for the 7807) and assembly drawings (figure 5-2 for the 7806 and figure 5-4 for the 7807) in the following pages are included in this manual FOR REFERENCE USE ONLY. These drawings may differ in some respects from the card you receive from Pro-Log. If you require the specific revision documents of the schematic and assembly for the board shipped, please use the information request form in the back of this manual to order them.

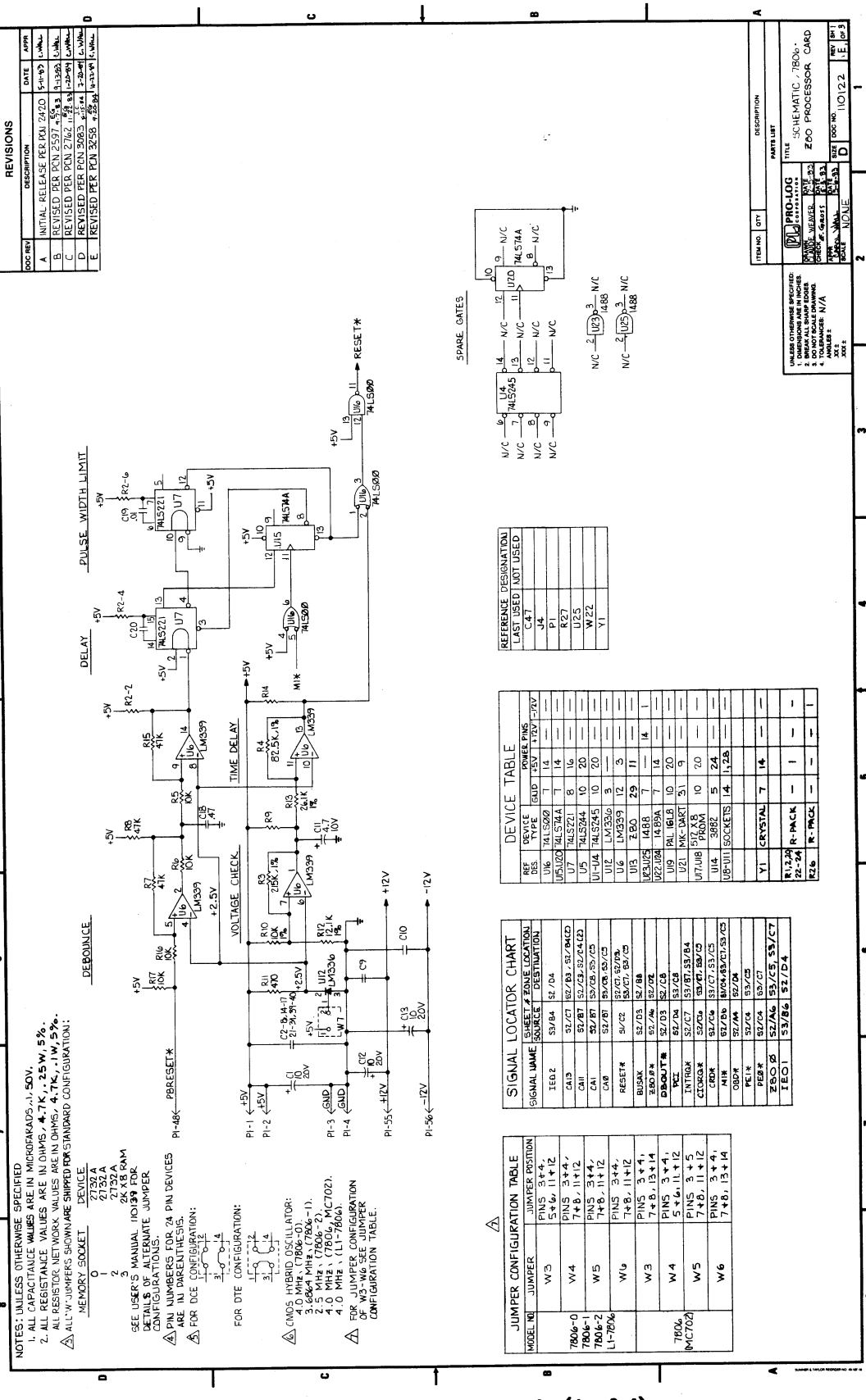


Figure 5-1. 7806 Schematic (1 of 4)
FOR REFERENCE ONLY

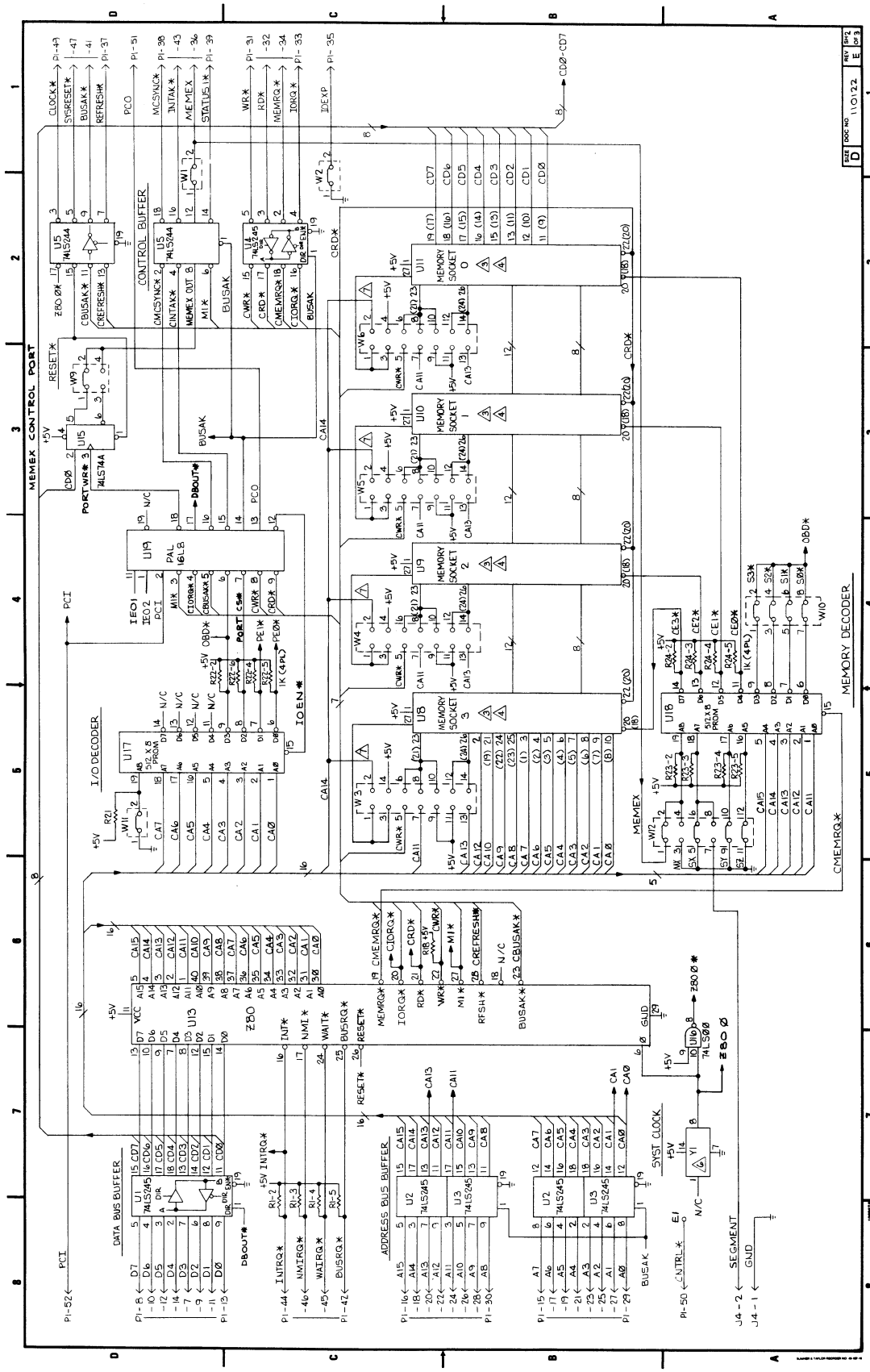


Figure 5-1. 7806 Schematic (2 of 4)
FOR REFERENCE ONLY

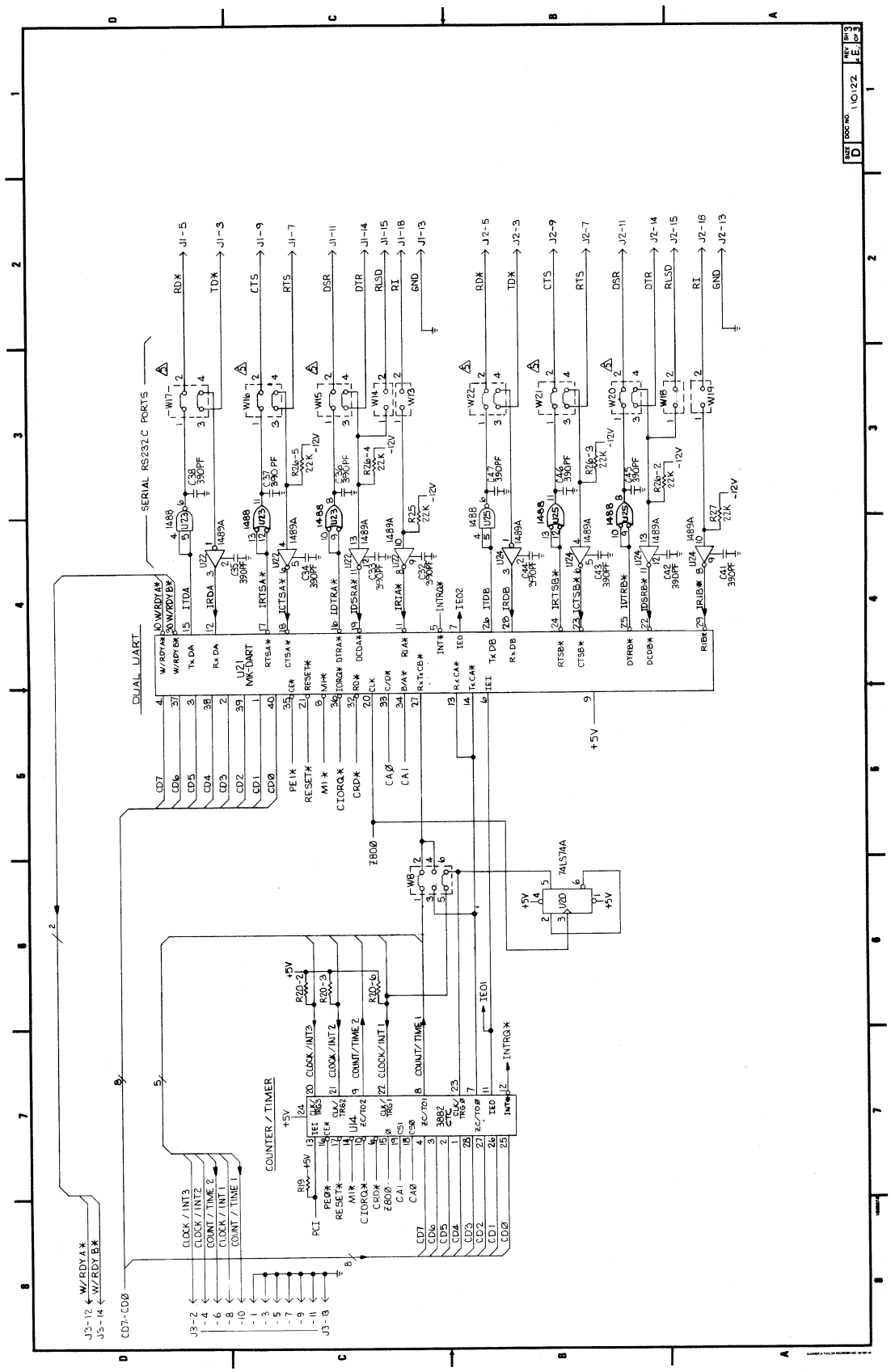


Figure 5-1. 7806 Schematic (3 of 4)
FOR REFERENCE ONLY

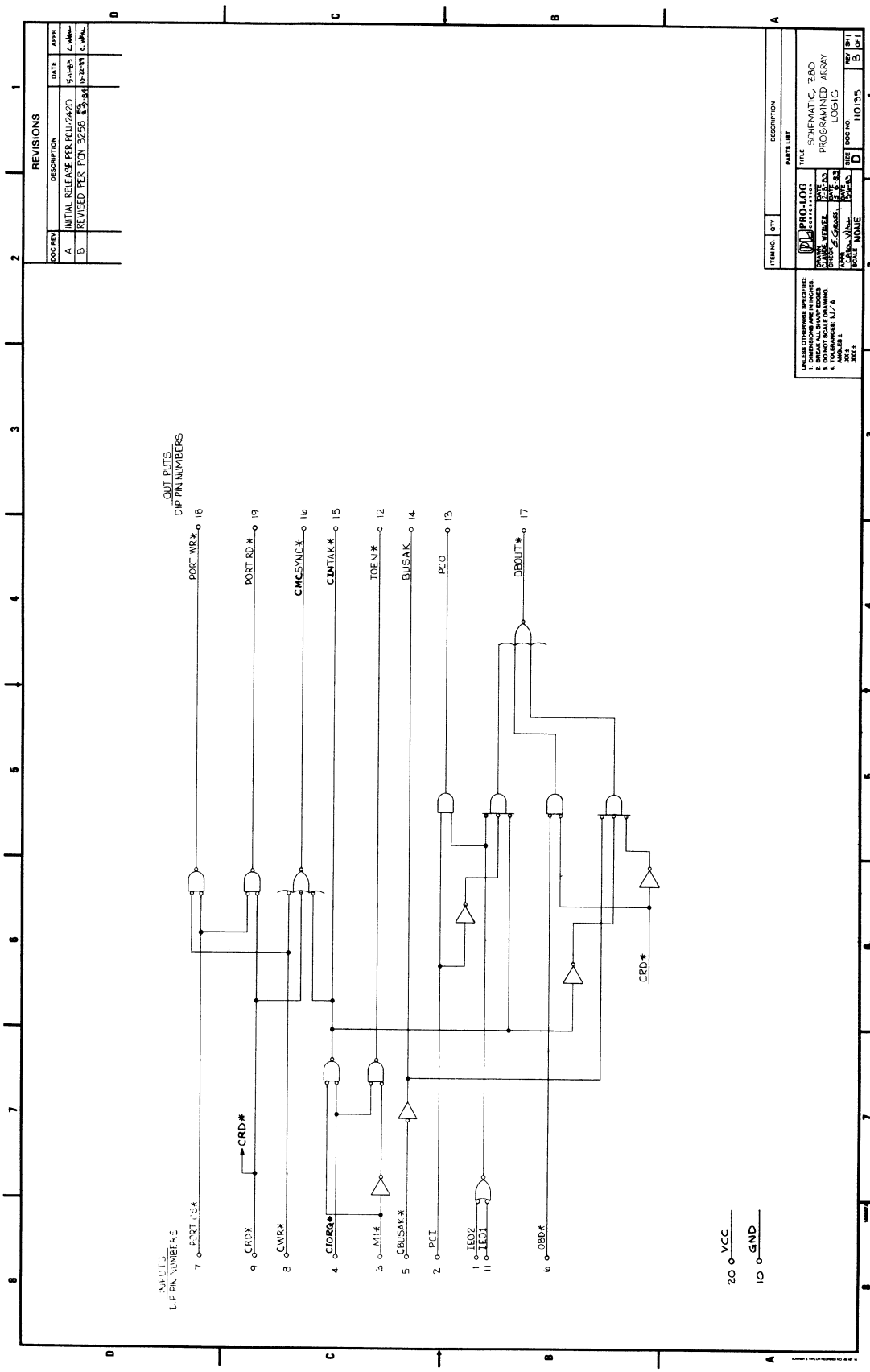


Figure 5-1. 7806 Schematic (4 of 4)
FOR REFERENCE ONLY

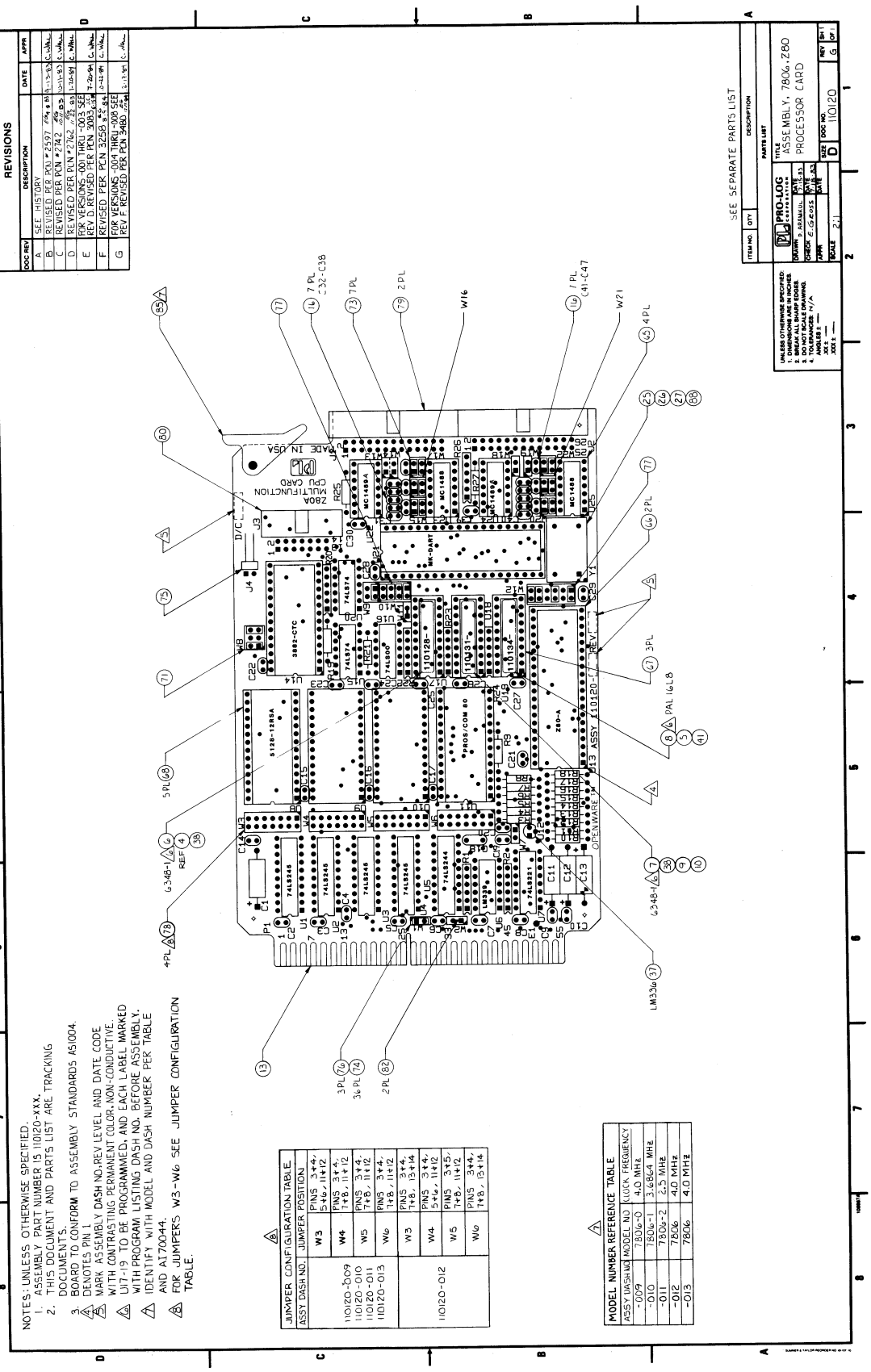


Figure 5-2. 7806 Assembly Drawing
FOR REFERENCE ONLY

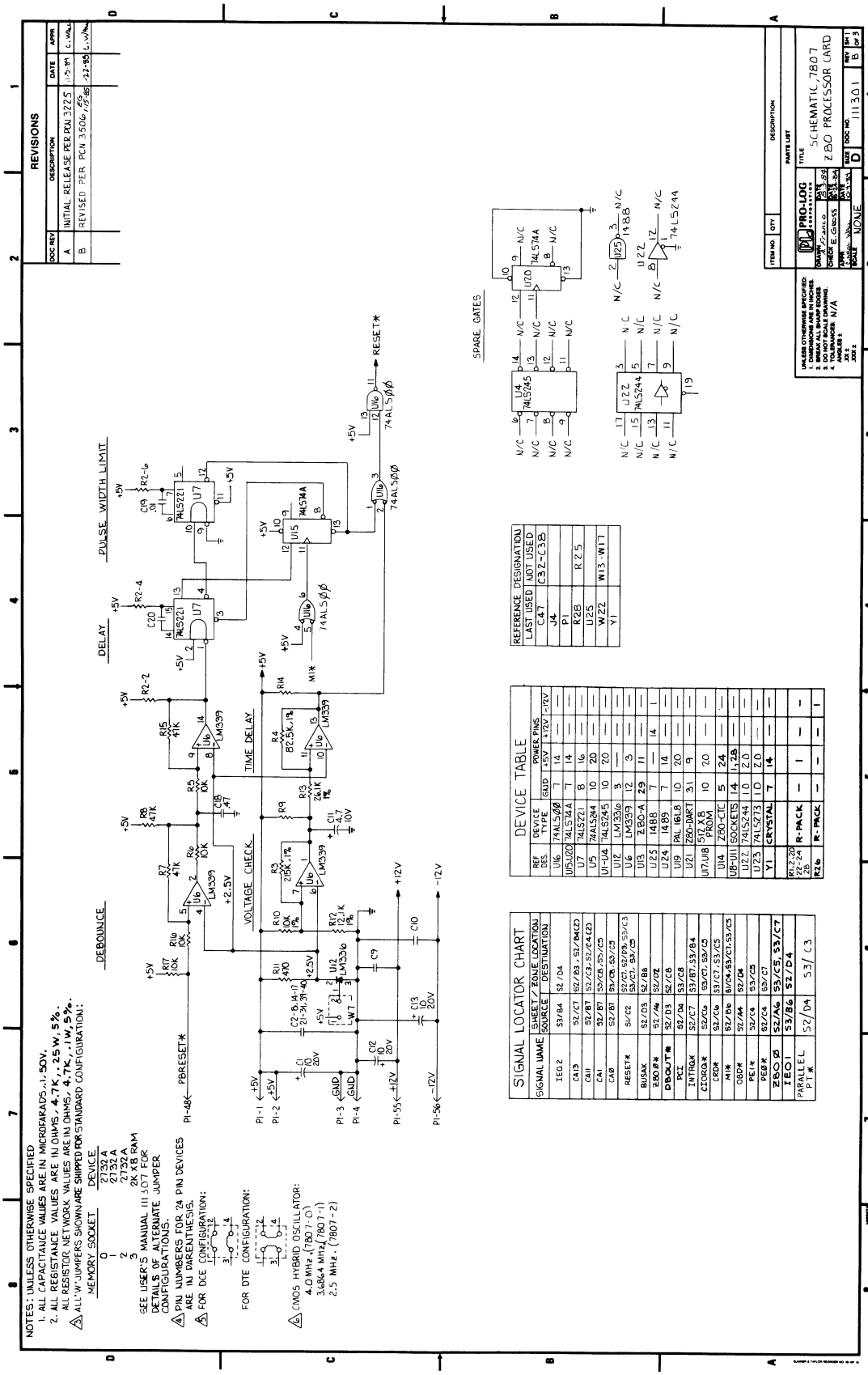
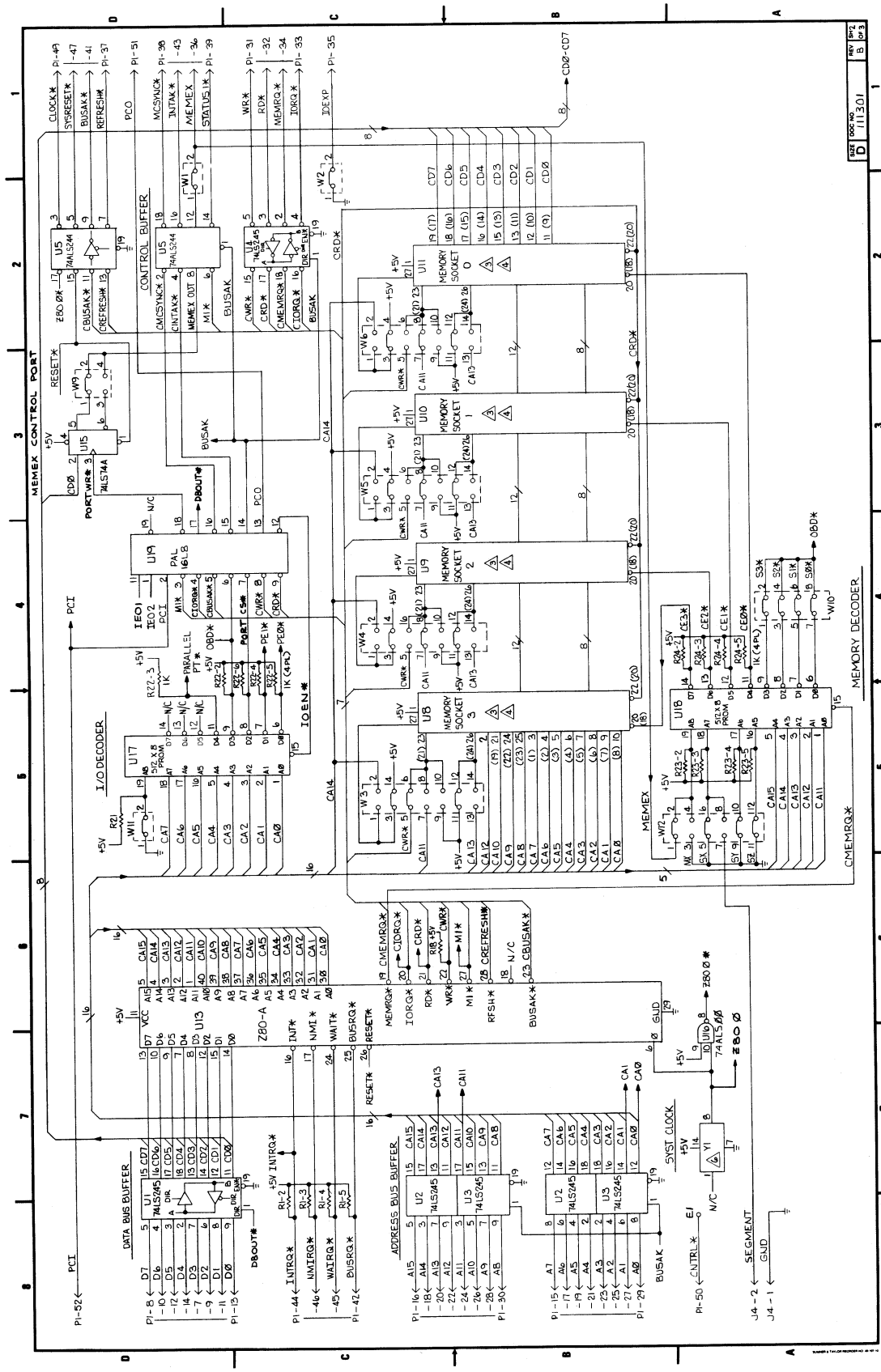


Figure 5-3. 7807 Schematic (1 of 4)
FOR REFERENCE ONLY


MEMEX CONTROL PORT
I/O DECODER
CONTROL BUFFER
DATA BUS BUFFER
ADDRESS BUS BUFFER
MEMORY DECODER
SYST CLOCK
SEGMENT
PARALLEL
PORT CS#
IOEN#
DBOUT#
INTRQ#
NMIRQ#
WAITRQ#
BUSRQ#
RESET#
CLOCK#
SYSRESET#
BUSAK#
REFRESH#
MCSYNC#
INTAK#
MEMEX
STATUS1#
WR#
RD#
MEMRQ#
IORQ#
IOEXP
CMEMRQ#
CIORQ#
CRD#
CWR#
M1#
RFSH#
CBUSAK#
CREFRESH#
CD0-CD7
MEMORY SOCKET
74LS244
74LS245
74LS374
74LS138
74LS32
74LS04
74LS00
74LS86
74LS74A
74LS08
74LS139
74LS157
74LS153
74LS257
74LS273
74LS175
74LS125
74LS14
74LS20
74LS21
74LS27
74LS30
74LS51
74LS86

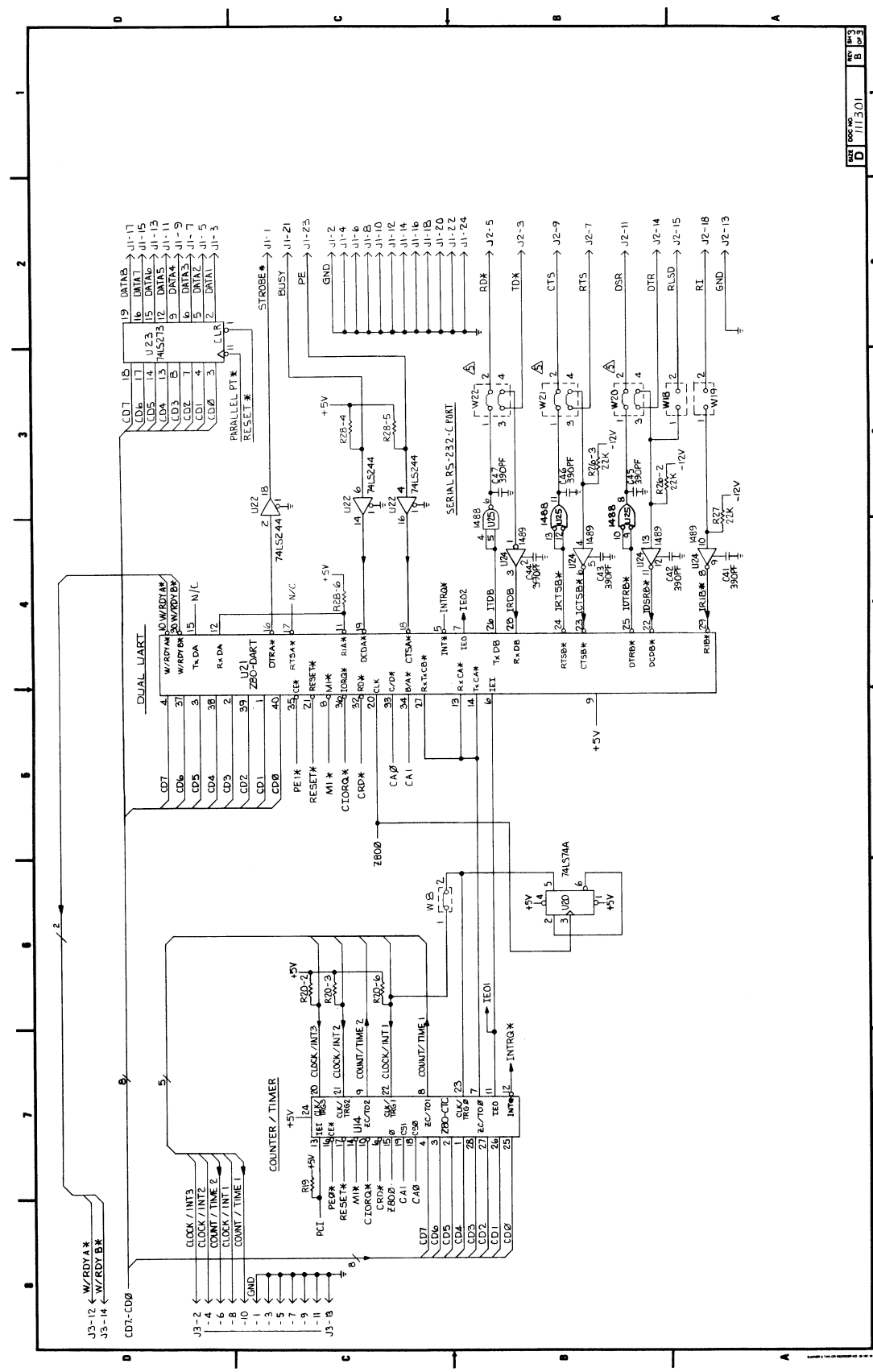


Figure 5-3. 7807 Schematic (3 of 4)
FOR REFERENCE ONLY

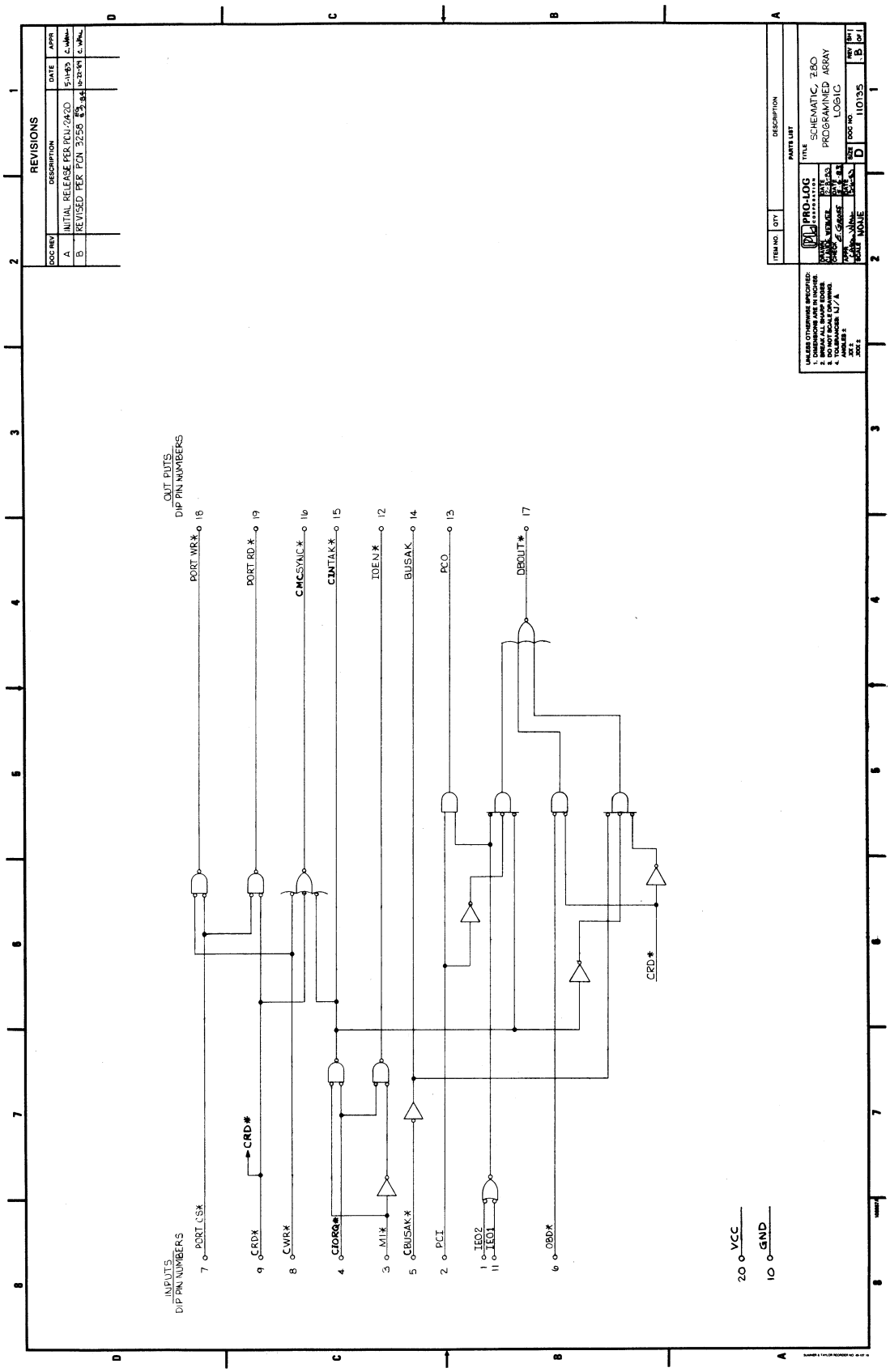


Figure 5-3. 7807 Schematic (4 of 4)
FOR REFERENCE ONLY

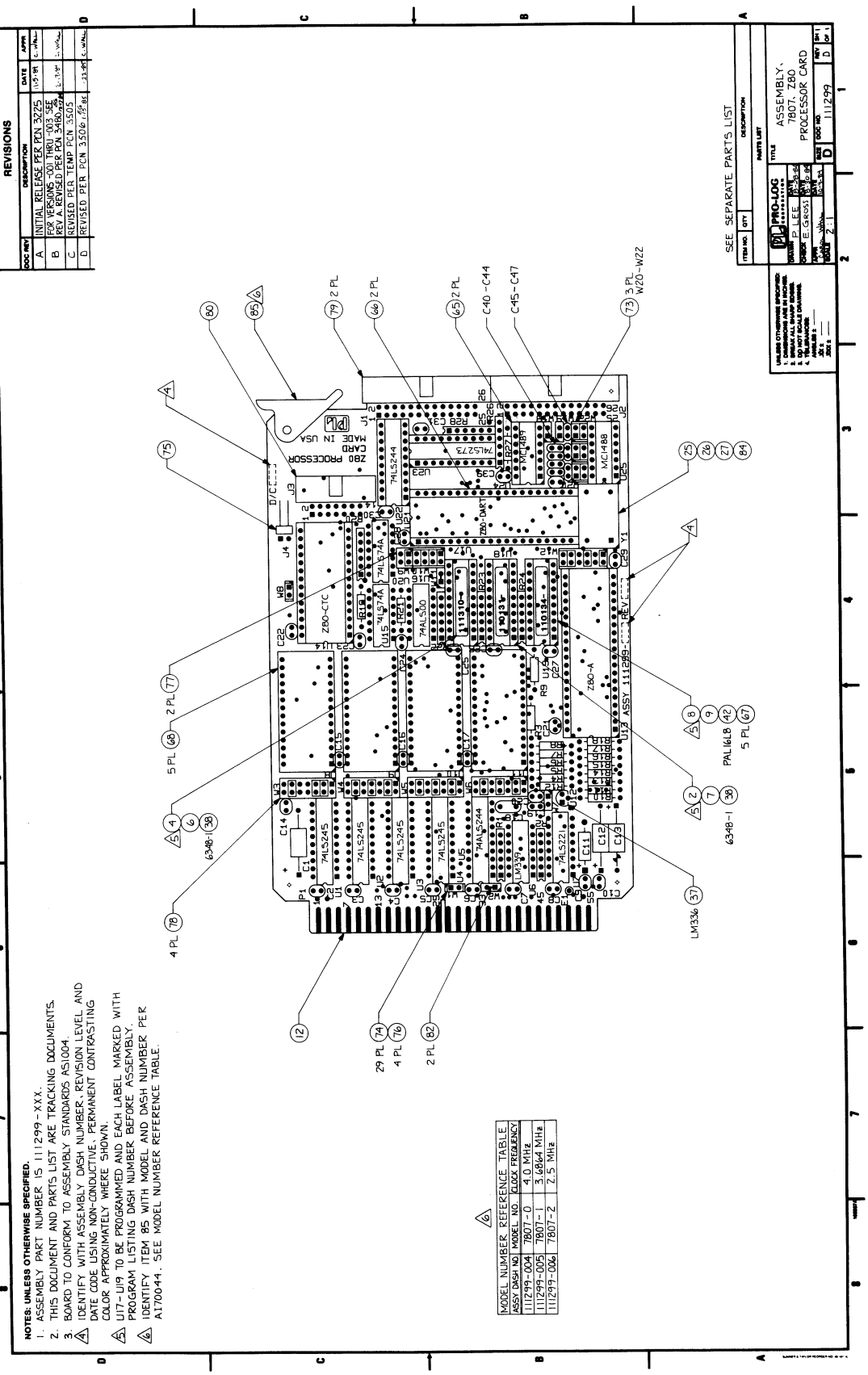


Figure 5-4. 7807 Assembly Drawing
FOR REFERENCE ONLY

RETURN FOR REPAIR PROCEDURES

Domestic Customers:

1. Call CUSTOMER SERVICE at (408) 372-4593 (in California) or (800) 538-9570 (outside of California) 7 AM to 3 PM PST.

2. Explain the problem and we may be able to solve it on the phone. If not, we will give you a Customer Return Order (CRO) number.

   Mark the CRO number on the shipping label, packing slip, and other paperwork accompanying the return. We cannot accept returns without a CRO.

3. Please be sure to enclose a packing slip with the CRO number, the equipment serial number if applicable, and the reason for return. Also enclose the name and telephone number of the person we should contact (preferably the user).

4. Package the equipment in a solid cardboard box secured with packing material.

CAUTION: Loose MOS integrated circuits, or any product containing CMOS integrated circuits, must be protected from electrostatic discharge during handling. Use conductive foam pads or conductive plastic bags, and never place MOS or CMOS circuitry in contact with Styrofoam materials.

5. Ship prepaid and insured to:

   Pro-Log Corporation
   2411 Garden Road
   Monterey, CA 93940
   Reference CRO #_____.

International Customers:

Equipment repair is handled by your local Pro-Log distributor. If you need to contact Pro-Log, the factory can be reached at any time by TWX at 910-360-7082.

LIMITED WARRANTY:

Seller warrants that the articles furnished hereunder are free from defects in material and workmanship and perform to applicable, published Pro-Log specifications for two years from date of shipment. This warranty is in lieu of any other warranty expressed or implied. In no event will Seller be liable for special or consequential damages as a result of any alleged breach of this warranty provision. This liability of Seller hereunder shall be limited to replacing or repairing, at its option, any defective units which are returned F.O.B. Seller's plant. Equipment or parts which have been subject to abuse, misuse, accident, alteration, neglect, unauthorized repair, or installation are not covered by warranty. Seller shall have the right of final determination as to the existence and cause of defect. As to items repaired or replaced, the warranty shall continue in effect for the remainder of the warranty period, or for ninety (90) days following date of shipment by Seller of the repaired or replaced part whichever period is longer. No liability is assumed for expendable items such as lamps and fuses. No warranty is made with respect to custom equipment or products produced to buyer's specifications except as specifically stated in writing by Seller and contained in the contract.

INTRODUCTION

This appendix is a compilation of tables and figures which appear elsewhere in the manual. They are presented here for quick access to information you will need to refer to frequently. These figures can also be found at the figure number included at the end of the title of each figure, along with additional information.

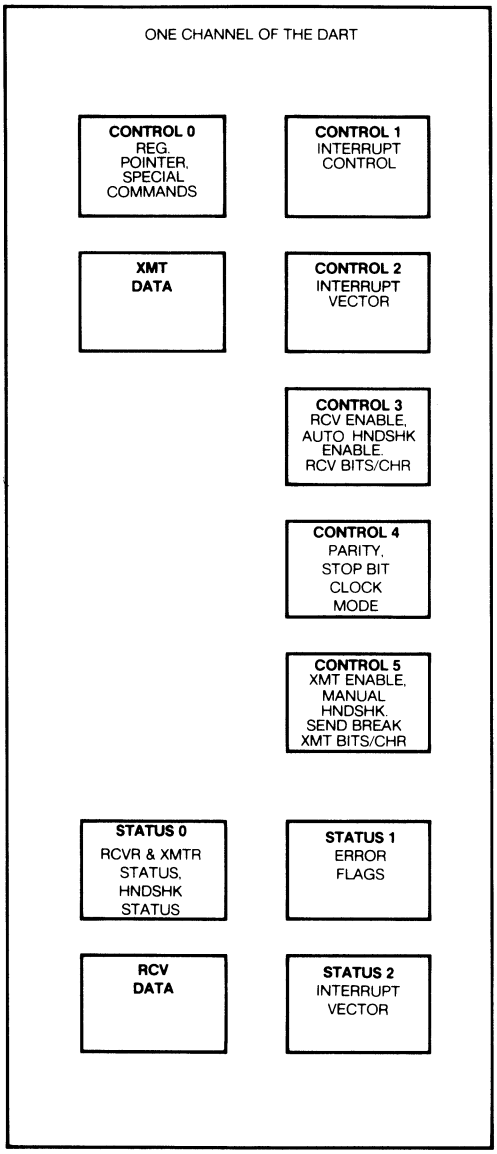


Figure A-1. DART Register Functions, General (7806/7 Figure 3-3)

REG	BIT 7	6	5	4	3	2	1	0
RC0			SPECIAL COMMANDS[1]			REGISTER POINTER		
RC1	W/Ready Enable	W/Ready acts as Ready	W/Ready on RCV	RCVR INT Control[2]		Status Affects Vector †	XMTR INT Enable	Hndshk INT Enable
RC2	INTERRUPT VECTOR †							
RC3	RCVR Bits/ Character[3]		Auto Handshake					RCVR Enable
RC4	Clock Divider[4]				Stop Bits[5]		Parity Even	Parity Enable
RC5	DTR/DSR STROBE* (7807 CH A)	XMTR Bits/Character[3]		Send Break	XMTR Enable		RTS/CTS	
RS0	Break		CTS/RTS PE (7807 CH A)	RI	DCD/DSR/DTR BUSY (7807 CH A)	XMTR Buffer Empty	INT Pending §	RCV Char. Available
RS1		Framing Error	RCVR Overrun	Parity Error				Thru Sending
RS2	INTERRUPT VECTOR[6] †							

† Channel B only; § Channel A only

[1] 000 NOP
001 Invalid
010 Reset handshake interrupt
011 Channel reset
100 Re-enable INT on first RCVD character
101 Reset pending XMTR INT
110 Error reset
111 Return from INT*

[2] 00 RCVR INT disabled
01 RCVR INT on first character
10 RCVR INT on every character, parity affects vector
11 RCVR INT on every character, parity does not affect vector

[3] 00 5 Bits/character or less.
01 7 Bits/character
10 6 Bits/character
11 8 Bits/character

When transmitting 5 or less bits/character the following pattern must be followed:

D7	D6	D5	D4	D3	D2	D1	D0	
1	1	1	1	0	0	0	D	One Bit
1	1	1	0	0	0	D	D	Two Bits
1	1	0	0	0	D	D	D	Three Bits
1	0	0	0	D	D	D	D	Four Bits
0	0	0	D	D	D	D	D	Five Bits

D=DATA

[4] 00 Clock divided by 1
01 Clock divided by 16
10 Clock divided by 32
11 Clock divided by 64

[5] 00 Invalid
01 1 Stop Bits
10 1½ Stop Bits
11 2 Stop Bits

[6]* If the Status Affects Vector mode is enabled (RC1), bits 3, 2, and 1 of the interrupt vector will be:

	* B3	B2	B1	
	0	0	0	XMTR Buffer Empty
CH	0	0	1	Handshake Change
B	0	1	0	RCVD character ready
	0	1	1	Special RCV condition
	1	0	0	XMTR Buffer Empty
CH	1	0	1	Handshake Change
A	1	1	0	RCVD character ready
	1	1	1	Special RCV condition

*Channel B only; **Channel A only

Figure A-2. DART Register Functions, Detailed (7806/7 Figure 3-4)

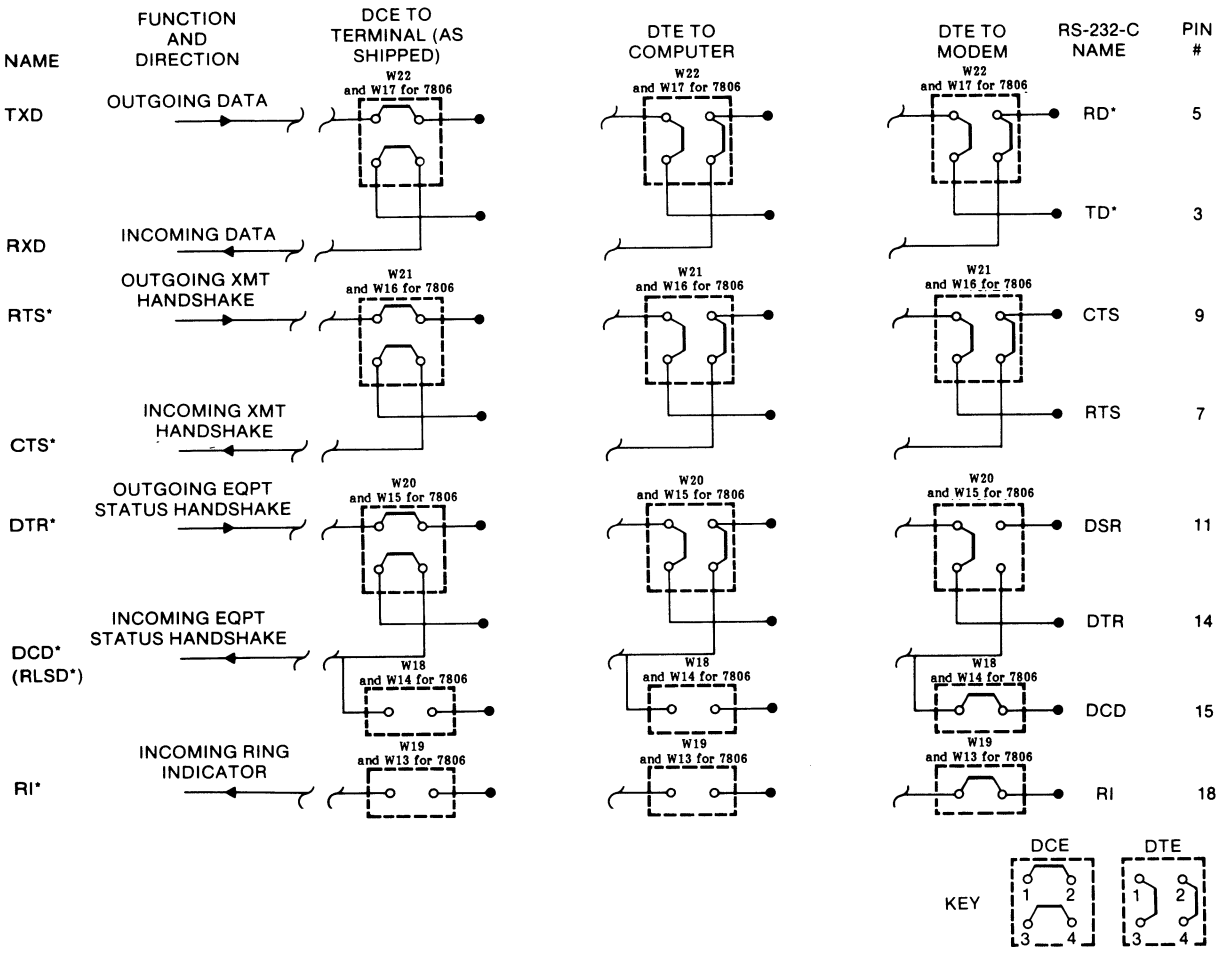


Figure A-3. Serial Port DCE/DTE Jumpers (7806/7), Figure 2-4)

BAUD RATE	CTC CLOCK DIVIDER	CTC COUNT DEC	CTC COUNT HEX	CTC MODE	DART CLOCK DIVIDER	ACTUAL BAUD RATE % ERROR
4 MHz (7806/7)						
9600	1	13	0D		16	+0.16
4800	1	26	1A		16	+0.16
2400	1	52	34		16	+0.16
1800	1	69	45		16	+0.64
1200	1	104	68	COUNTER	16	+0.16
600	1	208	D0		16	+0.16
300	16	52	34		16	+0.16
200	16	78	4E		16	+0.16
150	16	104	68		16	+0.16
134.5	16	116	74	TIMER	16	+0.15
110	16	142	8E		16	+0.03
75	16	208	D0		16	+0.16
50	16	78	4E		64	+0.16
3.6864 MHz (7806/7)						
38.4K	1	3	03		16	0
19.2K	1	6	06		16	0
9600	1	12	0C		16	0
4800	1	24	18		16	0
2400	1	48	30	COUNTER	16	0
1800	1	64	40		16	0
1200	1	96	60		16	0
600	1	192	C0		16	0
300	16	48	30		16	0
200	16	72	48		16	0
150	16	96	60		16	0
134.5	16	107	6B	TIMER	16	+0.06
110	16	131	83		16	-0.07
75	16	192	C0		16	0
50	16	72	48		64	0
2.5 MHz (7806/7)						
38.4K	1	2	2		16	+1.72
19.2K	1	4	4		16	+1.72
9600	1	8	8		16	+1.72
4800	1	10	16		16	+1.72
2400	1	21	33	COUNTER	16	-1.36
1800	1	2B	43		16	+0.94
1200	1	41	65		16	+0.16
600	1	82	130		16	+0.16
300	16	21	33		16	-1.36
200	16	31	49		16	-0.35
150	16	41	65		16	+0.16
134.5	16	49	73	TIMER	16	-0.54
110	16	59	89		16	-0.25
75	16	82	130		16	+0.16
50	16	C3	195		16	+0.16

NOTE: Counter mode: Baud rate = ((System clock ÷ 2) ÷ CTC count) ÷ DART clock divider
Timer mode: Baud rate = ((System clock ÷ CTC clock divider) ÷ CTC count) ÷ DART clock divider

Figure A-4. Baud Rate Selection (7806/7 Figure 3-5)

I/O PORT ADDR	OUTPUT FUNCTION	INPUT FUNCTION
F0	CTC CHANNEL 0 Control, Vector, and Load Registers	CTC CHANNEL 0 Current Count Register
F1	CTC CHANNEL 1 Control and Load Registers	CTC CHANNEL 1 Current Count Register
F2	CTC CHANNEL 2 Control and Load Registers	CTC CHANNEL 2 Current Count Register
F3	CTC CHANNEL 3 Control and Load Registers	CTC CHANNEL 3 Current Count Register
F4	DART CHANNEL A XMT Data	DART CHANNEL A RCV Data
F5	DART CHANNEL A Control Registers (Parallel Handshake)	DART CHANNEL A Status Registers (Parallel Handshake)
F6	DART CHANNEL B XMT Data	DART CHANNEL B RCV Data
F7	DART CHANNEL B Control Registers	DART CHANNEL B Status Registers
FD (7807 only)	8-BIT PARALLEL OUTPUT PORT	Not Used
FE	MEMEX CONTROL	Not Used

Figure A-5. I/O Port Functions (7806/7 Figure 3-1)

PRIORITY	DESCRIPTION	VECTOR
Highest		
1	CTC Channel 0	XXXX X000
2	CTC Channel 1	XXXX X010
3	CTC Channel 2	XXXX X100
4	CTC Channel 3	XXXX X110
5	DART Ch. A, RCV character or special RCV condition	XXXX 110X XXXX 111X
6	XMT Buffer Empty	XXXX 100X
7	Handshake Transition	XXXX 101X
8	DART Ch. B, RCV character or special RCV condition	XXXX 010X XXXX 011X
9	XMT Buffer Empty	XXXX 000X
10	Handshake Transition	XXXX 001X
Lowest		

Figure A-6. Interrupt Vectors and Priority (7806/7 Figure 3-6)

BIT							
7	6	5*	4	3*	2	1	0
ENABLE INTS	COUNT-ER/TIMER	CLOCK DIVIDER	CLOCK EDGE	TRIGGER	LOAD REG.	CHANNEL RESET	LOAD VECTOR

BIT	DESCRIPTION
7	1=Enable Interrupts
6	1=Counter Mode 0=Timer Mode
5*	1=Clock divided by 256 0=Clock divided by 16
4	1=External clock/trigger line is rising edge sensitive 0=Falling edge sensitive
3*	1=Timer started by external trigger 0=Timer starts after being loaded
2	1=Next byte to this channel is load register data
1	1=Channel stops; must be reloaded or re-initialized Momentary function
0	Channel 0 only: 0=byte to be loaded to vector register

*Timer mode only

Figure A-7. CTC Control Registers (7806/7 Figure 3-10)

NAME	DESCRIPTION	USE REQUIREMENTS
(INIT COMM)	Initialize Communications Interface (Entry Address 1D00 hex)	None
(STRT XMT A)	Transmit a block of data (CH A) (Entry Address 1DC0 hex)	Data must be in memory, data pointer in register pair HL, and number of bytes in register A
(STRT XMT B)	See above (CH B). (Entry Address 1DE0 hex)	See above
(RCV & ECHO A)	Receive and echo a character (CH A) (Entry Address 1E40 hex)	User must provide software to store and/or respond to the received character.
(RCV & ECHO B)	See above (CH B). (Entry Address 1E50 hex)	See above

Figure A-8. 7806 Main Subroutines Use Requirements (Figure 4-2)

NAME	DESCRIPTION	USE REQUIREMENTS
(INIT SER COMM)	Initialize Serial Communications Interface (Entry Address 1D00)	Interrupt vector page address at 1D1B
(INIT PAR PORT)	Initialize parallel (printer) interface (Entry Address 1D30)	None
(STRT XMT B)	Transmit a block of data (CH B) (Entry Address 1DE0)	Data must be in memory, data pointer in register pair HL, and number of bytes in register A
(PRINT DATA A)	Outputs a block of characters through parallel printer interface (Entry Address 1E20)	Data must be in memory, data pointer in register pair HL, and number of bytes in register A
(RCV & ECHO B)	Receive and echo a character (CH B) (Entry Address 1E50)	User must provide software to store and/or respond to the received character

Figure A-9. 7807 Main Subroutine Use Requirements (Figure 4-24)

Cut Here

FOLD

FOLD

PRO-LOG® CORPORATION

Yes, I'm interested in Pro-Log.

Please send me more information about your:

- ☐ 7000 Series STD BUS Cards
- ☐ STD BUS Packaging Solutions
- ☐ STD BUS Subsystem
- ☐ STD BUS Systems
- ☐ Designer Software
- ☐ Design Courses and Seminars
- ☐ Programmer Product Line
- ☐ Other _____

- ☐ I am currently using Pro-Log equipment.
- ☐ Add my name to your mailing list.
- ☐ Please have a Sales Representative contact me.

Reader Comments: _____

Name _____ City _____

Company _____ State _____ Zip _____

Address _____ Telephone _____

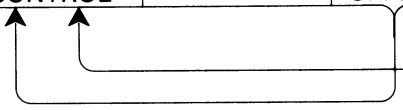
PLACE
POSTAGE
HERE

PRO-LOG®
CORPORATION
2411 GARDEN ROAD
MONTEREY, CA 93940

Attn: Literature Department

Notes

HEX				MSD	p=1	8	9	A	B	C	D	E	F
					p=0	0	1	2	3	4	5	6	7
	BITS				b8	p	p	p	p	p	p	p	p
					b7	0	0	0	0	1	1	1	1
					b6	0	0	1	1	0	0	1	1
LSD	b4	b3	b2	b5 / b1		0	1	0	1	0	1	0	1
0	0	0	0	0		NUL	DLE	SP	0	@	P	‘	p
1	0	0	0	1		SOH	DC1	!	1	A	Q	a	q
2	0	0	1	0		STX	DC2	"	2	B	R	b	r
3	0	0	1	1		ETX	DC3	#	3	C	S	c	s
4	0	1	0	0		EOT	DC4	$	4	D	T	d	t
5	0	1	0	1		ENQ	NAK	%	5	E	U	e	u
6	0	1	1	0		ACK	SYN	&	6	F	V	f	v
7	0	1	1	1		BEL	ETB	’	7	G	W	g	w
8	1	0	0	0		BS	CAN	(	8	H	X	h	x
9	1	0	0	1		HT	EM	)	9	I	Y	i	y
A	1	0	1	0		LF	SUB	*	:	J	Z	j	z
B	1	0	1	1		VT	ESC	+	;	K	[	k	{
C	1	1	0	0		FF	FS	,	<	L	\	l	|
D	1	1	0	1		CR	GS	—	=	M	]	m	}
E	1	1	1	0		SO	RS	.	>	N	^	n	~
F	1	1	1	1		SI	US	/	?	O	_	o	DEL
						CONTROL				UPPER CASE		LOWER CASE	



CONTROL CHARACTERS AND STANDARD KEYBOARD CONTROL SHIFT CHARACTERS

CTRL	@	NUL	Null	CTRL	K	VT	Vertical Tabulation	CTRL	V	SYN	Synchronous Idle
CTRL	A	SOH	Start of Heading	CTRL	L	FF	Form Feed	CTRL	W	ETB	End of Transmission Block
CTRL	B	STX	Start of Text	CTRL	M	CR	Carriage Return	CTRL	X	CAN	Cancel
CTRL	C	ETX	End of Text	CTRL	N	SO	Shift Out	CTRL	Y	EM	End of Medium
CTRL	D	EOT	End of Transmission	CTRL	O	SI	Shift In	CTRL	Z	SUB	Substitute
CTRL	E	ENQ	Enquiry	CTRL	P	DLE	Data Link Escape	CTRL	[	ESC	Escape
CTRL	F	ACK	Acknowledge	CTRL	Q	DC1	Device Control 1	CTRL	\	FS	File Separator
CTRL	G	BEL	Bell	CTRL	R	DC2	Device Control 2	CTRL	]	GS	Group Separator
CTRL	H	BS	Backspace	CTRL	S	DC3	Device Control 3	CTRL	^	RS	Record Separator
CTRL	I	HT	Horizontal Tabulation	CTRL	T	DC4	Device Control 4	CTRL	_	US	Unit Separator
CTRL	J	LF	Line Feed	CTRL	U	NAK	Negative Acknowledge			DEL	Delete

NOTE: The table above lists the keyboard character to be pressed with the Control key, the mnemonic for the function to be performed, and the description of the function. While the keyboard character is listed as uppercase, either uppercase or lowercase characters may be used.

ASCII Code Assignments

USER'S MANUAL

PRO-LOG® CORPORATION
2560 Garden Road, Monterey, California 93940
Phone: (408) 372-4593, (800) 538-9570
Telex: 171879, TWX: 910-360-7082

110139-002D 3K 12/85